KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Improving the robustness of a railway system in large and complex station areas

Thijs Dewilde

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

December 2014

Improving the robustness of a railway system in large and complex station areas

Thijs DEWILDE

Supervisory Committee:
Prof. dr. ir. P. Verbaeten, chair
Prof. dr. ir. P. Vansteenwegen, supervisor
Prof. dr. ir. D. Cattrysse, supervisor
Prof. dr. L. Suhl, co-supervisor
(University of Paderborn)
dr. ir. F. Corman
Prof. dr. ir. G. Vanden Berghe
Prof. dr. L. Kroon
(Erasmus University Rotterdam)
Prof. dr. K. Sörensen
(University of Antwerp)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

December 2014

D/2014/7515/148
ISBN 978-94-6018-924-1

Dankwoord

Het is allemaal begonnen met het idee dat doctoreren toch niets was voor mij. Dat was echter buiten dit interessante project gerekend waardoor dit idee zijn eigen treinritje ging maken. Dat mijn doctoraatsrit alle stopseinen, storingen en hoogteverschillen heeft overwonnen en toch nog plaats liet voor successen zowel op persoonlijk als op onderzoeksgerelateerd vlak is te danken aan verschillende mensen.

Ten eerste zijn dat mijn promotoren Pieter Vansteenwegen en Dirk Cattrysse. Pieter, hoe vaak ben ik niet komen aankloppen met de vraag of je geen vijf seconden tijd had voor mij. Die vijf seconden, die vaak meer dan 30 minuten bleken, hebben er echt wel voor gezorgd dat mijn treintje op de rails bleef. Dirk, met een vrolijke noot en een bewonderenswaardige dosis inzicht heb je een grote bijdrage geleverd aan het behalen van de eindmeet.

Daarnaast wens ik de leden van de examencommissie te bedanken: Leena Suhl, Greet Vanden Berghe, Kenneth Sörsensen, Francesco Corman, Leo Kroon en Pierre Verbaeten. Via interessante gesprekken op conferenties, constructieve feedback bij mijn werk en nuttige discussies waren jullie uitverkoren pijlers om mijn spoorwegbrug te ondersteunen.

Dit onderzoek is tot stand gekomen in nauwe samenwerking met de mensen van Infrabel en in het bijzonder Sabine Verboven en Eric Vercauteren. Als eerste aanspreekpunt wisten jullie elke vraag of elk probleem zelf grotendeels op te lossen of telkens de juiste mensen te vinden die ons verder konden helpen. Dankzij jullie lagen de wissels telkens in de juiste positie.

Ik ben ook het agentschap voor “Innovatie door Wetenschap en Technologie” (IWT) dankbaar voor de financiële ondersteuning die het mogelijk heeft gemaakt mijn kennis te vergroten en te delen op internationaal niveau en mij de mogelijkheid heeft gegeven om mijn doctoraat op een goede manier te beëindigen.

Mijn periode op het CIB en op de afdeling IR18 van de UGent heb ik altijd plezant gevonden. De kantoorgenootjes Reginald, Lorena, Corrinne, Ali en Nagham wil ik bedanken voor de vele babbels ter ontspanning, de Sofies voor het samenwerken en alle anderen voor de “koffiepauzes” waar ik mijn soepje kon drinken, de gezellige lunches of andere werkonderbrekingen. Daarnaast mag ik de mensen van het secretariaat niet vergeten. Een wijs man gaf me ooit de raad hen te vriend te houden want ik zou ze vaak nodig hebben. En inderdaad, voor veel kleine en grote zaken staan jullie telkens klaar om het over te nemen en alles goed te regelen zodat mijn treintje verder kan rijden. Bedankt daarvoor.

Voor sfeer en gezelligheid buiten de werkuren zorgden de vele afspraakjes met vrienden. Voor huisbezoeken, etentjes, een reis of uitstapjes, altijd stond er wel iemand paraat om onze sociale agenda te vullen en de bekommernissen van het doctoraat even naar de achtergrond te verdringen. Ook al zijn die zorgen nu van de baan, laat al die afspraken in jullie agenda maar staan want ik kijk er al naar uit.

Naast al deze mensen wil ik ook mijn ouders, familie en schoonfamilie bedanken. In eerste instantie wil ik mijn mama en papa bedanken voor de kansen die ze mij gegeven hebben. Mijn keuze om op Erasmus te gaan werd ten volle ondersteund, op kot gaan was geen probleem en als we dan in het weekend thuis waren, stonden jullie altijd voor ons klaar. Dat het met de Dewildes aan tafel altijd plezant en nooit stil is, is grotendeels aan jullie te danken. Ook aan die andere lawaaierige sfeermakers: Wouter, Tine en Brecht, merci om het pad te effenen voor jullie kleine broer, hem vanop de juiste afstand wat te laten klungelen en toch ten gepaste tijde in te grijpen. Ook mijn schoonouders wil ik bedanken, samen met Michiel en Lise zorgen jullie er voor dat we met plezier naar Bazel afzakken.

Aan het eind wil ik ook Eline bedanken. Men zegt altijd dat je het beste voor het einde moet sparen. Wel, ik heb me bij het schrijven van dit dankwoord al ettelijke malen moeten inhouden want quasi overal speel je wel een rol. Nog voor ik het zelf besefte, wist jij al dat dat spoorwegproject mijn toekomst zou worden. De voorbije jaren heb ik je te pas en te onpas lastig gevallen met werkgerelateerde zaken of “interessante” treinweetjes. Bij het schrijven van mijn doctoraat was jij het die mijn gedrag en afwezigheid moest verdragen en als we dan eens samen de deur uit gingen, was jij degene die de afspraakjes maakte en ze ook bijhield. Je loopt al sinds het begin van mijn Leuvense periode aan mijn zijde en ook al wordt Leuven nu verleden tijd voor mij, jij blijft, en dat maakt mij gelukkig.

Bedankt.

Abstract

At the time where sustainable mobility is a hot topic, efficiency is high up on the agendas and passenger service is more important than ever. This makes robustness one of the key elements to avoid staying behind with an unreliable railway system and low punctuality numbers. Therefore, the Belgian railway infrastructure manager Infrabel raised the question to develop principles and techniques that are applicable on the tactical level and make the Belgian railway timetable more robust.

Inspired by the observation that bottlenecks, and more specifically large and complex station areas, are the main sources of delays, the main objective of this dissertation is to improve the robustness of a railway system in large and complex station areas.

First of all, the challenge was to detect the essence of what robustness is about. Since serving passengers is the main goal of a public railway system, robustness should also focus on this service towards the passengers. Considering their valuation of travel time as key performance indicator, a railway system is defined to be robust against daily occurring, small disturbances when it optimizes this valuation of real travel time.

In order to obtain this robustness, an optimization process is designed that strives for spreading the use of resources in time. In this process, the routing of trains through the network is addressed, an integrated approach to exploit the potential of changing train orders and schedules is built, and the benefits of modifying the platform allocations can be evaluated.

The computational results prove that the developed algorithm is able to improve the robustness of the system with a considerable reduction of delays as a consequence. Other results are a reduced interaction between trains in switch zones, better spread arrival times, and more balanced platform occupations. All together, this enables a much more efficient use of the limited capacity in the various considered bottlenecks and under varying circumstances.

Beknopte samenvatting

In tijden waar duurzame mobiliteit een veelbesproken onderwerp is, efficiëntie hoog in het vaandel gedragen wordt en klantgerichtheid belangrijker is dan ooit, is robuustheid de sleutel om niet achter te blijven met een onbetrouwbaar spoorwegsysteem met lage stiptheidscijfers. Daarom lanceerde Infrabel, de infrastructuurbeheerder van het Belgische spoorwegnet, de vraag naar technieken en principes om de dienstregeling voor de Belgische spoorwegen robuuster te maken.

Geïnspireerd door de vaststelling dat de knelpunten, en meer specifiek de grote en complexe stationsomgevingen, de grootste bron van vertragingen zijn, is de hoofddoelstelling van dit doctoraatsonderzoek het verbeteren van de robuustheid van een spoorwegsysteem in grote en complexe stationsomgevingen.

De eerste uitdaging was het eenduidig en meetbaar definiëren van robuustheid. Aangezien reizigers de doelgroep vormen van het openbaar vervoer, gaat robuustheid in essentie over het verzekeren van de dienstverlening voor reizigers. Daarom wordt de waardering van de reistijd als belangrijkste kwaliteitsmaatstaf beschouwd. Op die manier kan gezegd worden dat een spoorwegsysteem robuust is als de totale waardering van de effectieve reistijd in de praktijk zo goed mogelijk is.

Om de robuustheid te verbeteren, is een algoritme ontworpen dat het gebruik van de spoorweginfrastructuur in de tijd spreidt. Daarbij worden de routes van de treinen doorheen het netwerk beschouwd, is een geïntegreerde aanpak ontwikkeld die het potentieel van veranderingen in de volgorde van treinen en in de dienstregeling benut, en worden de mogelijkheden van perronwijzigingen bestudeerd.

De bekomen resultaten tonen aan dat het ontwikkelde algoritme in staat is om de robuustheid te verbeteren. Dit resulteert in een aanzienlijke afname van vertragingen. Andere verbeteringen zijn een gedaalde interactie tussen treinen in de wisselzones, beter gespreide aankomsttijden en meer evenwichtig verdeelde perronbezettingen. Alles samen laat dit het spoorwegsysteem toe om veel efficiënter om te gaan met de beperkte capaciteit in de verschillende knelpunten en in verscheidene omstandigheden.

Abbreviations

CPF cycle periodicity formulation.

DONS designer of network schedules.

LP linear programming.

LUKS *Leistungs-Untersuchungen für Knoten und Strecken* (Analysis of lines and junctions).

MILP mixed integer linear problem.

MIP mixed integer problem.

NPP node packing problem.

NSC North-South connection.

NTT nominal travel time.

PESP periodic event scheduling problem.

RCL restricted candidate list.

RTS running time supplement.

RWTT real weighted travel time.

$\mathbf{RWTT_{ext}}$ real weighted travel time extension.

$\mathbf{RWTT_{norm}}$ normalized real weighted travel time.

SSHR sum of the shortest headway reciprocals.

TPP train platforming problem.

TRP train routing problem.

TTP train timetabling problem.

List of symbols

Bold symbols are also used in later chapters.

Chapter 2	
H	Period of a periodic timetable
π_i	PESP decision variable for the event time of event i
l_{ij}	Lower bound for the time between events i and j
u_{ij}	Upper bound for the time between events i and j
k_{ij}	PESP decision variable representing the phase shift
x_{ij}	CPF decision variable for the process time
w_{ij}	Number of passengers that start their journey with event i and end it with event j
z^*	Optimal objective function value of the nominal solution
δ	Light robustness parameter
Chapter 3	
D	Set of days (d) to compute the RWTT
P	Population of passengers (p)
E	Set of passenger events (e)
E_p	Set of events for passenger p
$\boldsymbol{S}$	Set of stations ($\boldsymbol{s}$)
$\boldsymbol{T}$	Set of trains ($\boldsymbol{t}$)
$\hat{\boldsymbol{D}}$	Real average delay
$\left(\boldsymbol{E}_{\boldsymbol{\lvert T\rvert/2}},\, \boldsymbol{P}^{\mathbf{(0.5)}}_{\boldsymbol{3\lvert T\rvert/4}}\right)$	Description of a delay scenario
$\boldsymbol{E}_{\boldsymbol{\lvert T\rvert/2}}$	Indication that half of the trains ($\lvert T\rvert/2$) get an exponential input delay for the corresponding action
$\boldsymbol{P}^{\mathbf{(0.5)}}_{\boldsymbol{3\lvert T\rvert/4}}$	Indication that three-quarters of the trains gets a fixed input delay of 0.5 minutes for the corresponding action
T_{Qi}	Timetable for the i^{th} quarter of 2010
$\mathrm{T}_{i\%}$	The 2010 timetable with $i\%$ supplements included

Chapter 4

$\boldsymbol{R}$	Set of routes ($\boldsymbol{r}$)
$\boldsymbol{R_t}$	Set of routes for train t
$\boldsymbol{T} \times \boldsymbol{R}$	Set of trainroutes $\boldsymbol{(t,r)}$
$\boldsymbol{BS}$	Set of block sections ($\boldsymbol{bs}$)
$\boldsymbol{B^{bs}_{(t,r_t),(t',r_{t'})}}$	Minimum time span between trainroutes (t, r_t) and $(t', r_{t'})$ in block section bs
$\boldsymbol{B_{(t,r_t),(t',r_{t'})}}$	Minimum time span between trainroutes (t, r_t) and $(t', r_{t'})$
$\boldsymbol{C_{(t,r_t),(t',r_{t'})}}$	Spreading cost of $B_{(t,r_t),(t',r_{t'})}$
bs^*	Critical block section
$\boldsymbol{iter}^{\max}$	Parameter for the stop criterion of the algorithm
$\boldsymbol{B}^{\max}$	Parameter that denotes the shortest duration of a time span that is considered insensitive to conflicts
σ	A signal
$dist(\sigma, \sigma')$	Distance between signals σ and σ'
$v_t^{\max}$	Maximally allowed speed of train t

Chapter 5

L^{bs}	Set of links $l = (i, j)$ within block section bs
l_r	A link on route r
r_l	A route that uses link l
$\leftrightarrow_{bs}$	Indication of compatible routes in bs
BL_r^{bs}	Blocked link set of route r within bs
BL_r	Blocked link set of route r
N^{bs}	Set of nodes within bs
N_{in}^{bs}	Set of nodes within bs different from signals
$E(i)$	Set of links surrounding node i
$E^-(i)$	Set of incoming links of node i
$E^+(i)$	Set of outgoing links of node i
$\preccurlyeq$	Indication of (train-)route dominance $x \preccurlyeq y \Leftrightarrow x$ dominates y
$Dom_{(t,r)}$	Dominance set of trainroute (t, r)
$\boldsymbol{x_{(t,r)}}$	Binary decision variable for the TRP that is 1 if (t, r) is part of the routing solution
$y_{(t,r),(t',r')}$	Continuous decision variable for the TRP that is 1 if (t, r) and (t', r') are part of the routing solution
$\boldsymbol{z_{(t,r)}}$	Continuous decision variable for the TRP that models the spreading cost of (t, r)
α	Parameter to compare the Kaufman-TRP model with the approach of Caimi

Chapter 6

δ_t	Size of the time shift for train t
ϑ	Counter of the tabu search algorithm
$\varepsilon_{\text{step}}$	Parameter that determines the width of the RCL interval
$\vartheta^{\max}$	Parameter for the upper bound of a time span to become active
$\boldsymbol{B_{t,t'}}$	Minimum time span between trains t and t'
RCL^{UB}	Upper bound the RCL interval
$t + \delta_t$	Train t after the shift of t with δ_t
(t, δ_t)	Shiftmove of t with δ_t
(t, t')	Active pair of trains
$\delta_{\min}$	Parameter that bounds the largest time shift (< 0)
$\delta^{\max}$	Parameter that bounds the largest time shift (> 0)
$a..(b)..c$	Indication of the elements $a, a+b, a+2b, \ldots, a+ib$, with i the largest integer for which $a + ib \leq c$
ST	Set of shifted trains
υ	Random integer that is drawn from the uniform distribution
$CS^{\max}$	Parameter indicating the maximal number of shift moves in combined shift
$B^k_{t,t'}$	Minimum time span between trains t and t' after the k^{th} shift of combined shift
δ^k	Size of the k^{th} shift
ST^k	Set of trains that are shifted in the k^{th} shift
$C^k_{t,t'}$	Spreading cost of $B^k_{t,t'}$
$\underline{t}$	Train closest to t that precedes t
$\overline{t'}$	Train closest to t' that succeeds t'
δ_{step}	Stepsize of the shiftsize updates during order swap
δ^{tot}	Size of the total cumulative shift for an order swap

Chapter 7

P	Set of platforms (p)
p^{cur}	Current platform of a train
p^{new}	Indication of the new platform of a train
$B^s_{t,t'}$	Smallest time span between trains t and t' in station s or at the neighboring grids of s
B^{margin}	Parameter for the time span margin between two trains
$B^{\text{margin},s}$	Parameter for the time span margin between two trains in station s or at the neighboring grids of s
$\Gamma^i_{t,s}$	Number of trains that interact with train t at station s
Γ^{margin}	Parameter that bounds the difference between the maximum number of intersecting routes and $\Gamma^i_{t,s}$

$\Psi^1_{s,p}$	Total occupation time of platform p in station s
$\Psi^2_{s,p}$	Number of trains that use platform p in station s
Ψ_1^{margin}	Parameter to indicate the occupation rate margin
Ψ_2^{margin}	Parameter that bounds the difference in number of trains that visits a platform

Chapter 9

Δ_{dep}	Difference in departure time for a train due to the rescheduling actions
Δ_{arr}	Difference in arrival time for a train due to the rescheduling actions
$P_{(t,r)}$	Penalty of trainroute (t, r)
e^{orig}	Event time of event e in the original timetable
e^{up}	Event time of event e in the updated timetable
$[-x, x]$	Short representation of $e^{\text{up}} \in [e^{\text{orig}} - x, e^{\text{orig}} + x]$
ε	Symbol for the increasing time window bounds

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

En ook nu botsten die treinen geen enkele keer tegen elkaar, men moest een wiskundig monster zijn om een schema uit te dokteren waarbij dit getal treinen vlekkeloos door de kelder tjoekte.

And now, too, those trains didn't crash into each other once. You'd have to be a mathematical monster to work out a schedule to keep this many trains chugging perfectly round your cellar.

Dimitri Verhulst in *De helaasheid der dingen (The Misfortunates)*

Trains can have delays. This claim can feel like knocking on an open door. That these delays can be knocking on delays to other trains and that trains can face "traffic jams" is less straightforward. Limiting the impact of one delayed train on other trains and on the entire railway system, that is the topic of this dissertation. In the previous sentence, on purpose, a difference is made between *other trains* and *the entire railway system* because railways are more than trains; the main reason why trains ride are the passengers. Thus optimizing the entire railway system should not be done without considering the passengers.

Trains can have delays. Where do delays come from? Numerous reasons exist and they are similar to the reasons why one cannot predict his exact arrival time when traveling by car. The reaction speed at a traffic light, the acceleration rate of the car or of the cars in front of you, your driving behavior or that of others, the number of other cars, the number of people with the same origin or destination, etc., all these factors can influence your travel time. The same holds for trains. Although the number of trains and their origin and destination is

Part of this chapter is based on Dewilde et al. (2013).

known, the number of passengers that board or alight and the exact acceleration curve of that particular train unit with that particular train driver is not. Thus there will always be uncertainty about the exact arrival time of a train in a station and this can lead to delays.

Trains can have delays. Why does one delayed train influence other trains? Trains run on tracks and stopping aside to let a fast train pass by can only happen if there is a siding or in a station. If a takeover is not possible, the fast train gets stuck behind the slow train and arrives late at its next station. Passengers on that train who want to transfer to another train risk to miss their transfer unless their connecting train is deliberately delayed in order to guarantee the transfer. In this case, the connecting train can lose its time-slot during which its trajectory out of the station was reserved and thus hinder other trains which then get delayed. This procedure continues and is known as the knock-on effect.

Trains can have delays. Yes, but they can also have less delays. At least, according to the conclusions of this dissertation in which the aim is to decrease the knock-on of delays by making the railway system more robust.

1.1 The relevance of robustness for railway systems

Comparing the punctuality numbers of the Belgian passenger railways for the last few years[1], one sees that, despite a lot of effort to improve the performance, not much progress was measured. Belgium, once the first in continental Europe with an operating railway system, now lags behind its neighboring countries if it comes down to punctuality. In countries like the Netherlands and Germany, a lot of research to improve the performance has been done. In the state-of-the-art literature, numerous ideas are suggested and investigated, but often the passengers, who are the clients of a public railway system, are being forgotten. In order to improve the performance of the railway system, for which robustness is considered as a key performance indicator (Cacchiani et al. 2014; Schittenhelm 2011; Van Oort 2011), the Belgian railway infrastructure manager Infrabel launched a cooperation with KU Leuven and this dissertation is the first of hopefully many dissertations that result from this cooperation.

It is impossible to avoid all delays, but one can try to minimize them when constructing the timetable. This is where robustness comes into play. In short, robustness is the property of a system that describes the negative impact of disturbances on that system.

[1] Source: http://www.infrabel.be/en/about-infrabel/punctuality/reports, consulted in September 2014.

Of particular interest for this research is the North-South connection (NSC), the link between the stations Brussels North and Brussels Midi in Belgium which forms the center of the Belgian railway network and is used close to full capacity. Daily, Infrabel records a large amount of delays in and around the NSC and since it contains the busiest stations with respect to both the number of trains and passengers[2], a robust local timetable for the NSC is essential for a good global performance. Studying large and complex station areas and especially the Brussels' station area, will therefore cover a large part of this dissertation.

Not only the railways benefit from a more robust system, also the entire society. Being late for work, taking private transport since the public transport is unreliable, the stress caused by short transfer times, etc., all incur a cost for the society which will decrease if the railway system becomes more robust, and thus more reliable. As a consequence, more commuters will opt for the train instead of using their own car (Van Oort 2011). Together with the emerging concept of sustainable mobility, this creates a growing demand for more trains, and thus, a robust system is required to counter the larger vulnerability to delays if more trains are scheduled.

1.2 Scope

This research situates itself in the tactical level planning phase. More specifically, the timetable generation is studied. It is assumed that the infrastructure is known and that the origin, destination and stopping pattern of each train are fixed. The arrival and departure times for all trains are considered as being modifiable. The assignment of physical train units and a crew is typically done after the timetable generation step (Lusby et al. 2011a) and is left aside here. When a schedule is operated, real-time dispatching measures are needed to ensure the safety and smoothness of the railway traffic. Since this is the subject of another research field, this is not considered in this dissertation.

The demand for trains is larger during peak hours than during off-peak hours. Therefore, we focus on the busiest moment of the day which is the most vulnerable for delays. Throughout this entire research, only passenger trains and no freight trains are considered. This does not mean that freight trains are irrelevant, but is due to the fact that they normally do not run during peak hours on locations with scarce capacity. Anyway, incorporating freight trains in the model would be straightforward.

[2]Source: http://www.treintrambus.be/actueel/blog/1216-opstapcijfers.html, (in Dutch) consulted in September 2014.

The construction of a passenger railway timetable is done several months before it is put in operation. As a consequence, no accurate information is available about where and when delays will occur. This makes it very difficult to take irregular and large disturbances into account. It is much more efficient to handle these disturbances by dispatching actions than by changing the design of the timetable. In this research, different concepts are developed in order to improve the quality of a timetable. The timetable should be constructed in such a way that delays, which cannot be avoided completely, cause as less hinder and as less knock-on delays as possible. This is done in this research: improving the robustness in case of regular delays that originate typically at peak hours.

Punctuality numbers point out that most of the delays originate in large and complex station areas (Yuan 2006). Therefore, this research focuses on this kind of areas. In particular the station area of Brussels with the NSC is used as case study in this dissertation, together with another large and complex station area, Antwerp. Although Belgian case studies are used, the developed principles are generally applicable.

1.3 Research questions

The general objective of this research can be formulated as follows: **develop principles and techniques that make the Belgian railway timetable more robust**. This is the original question Infrabel posed at the start up of this research. It indicates that the objective is not about designing a complete new timetable but more about working on methods and procedures to improve the robustness of an already existing timetable. From the start, two things became clear: there is no formal definition of railway robustness, and when considering robustness, not only the timetable matters, but also the infrastructure and the infrastructure usage. Therefore, it is better to talk about robustness of a system instead of robustness of a timetable. In this dissertation, however, both are used and, therefore, it is important to keep in mind that robustness is not only related to the timetable but to the entire railway system. The lack of a formal definition of railway robustness gave rise to the first research question.

> **Research Question 1:** How to define the robustness of a railway system? What is the contribution of different elements in obtaining a robust railway system?

The answer to this question is given in Chapter 3. In literature, one can find various interpretations of what robustness is about. Based on an analysis of each interpretation, we formulate a new definition of railway robustness and discuss the advantages and disadvantages of this new definition. One of the advantages is that it becomes easy to measure and compare the robustness of different timetables. This is illustrated by measuring the robustness of the 2010 timetable for the whole Belgian railway network.

> **Research Question 2:** How to deal with the limited capacity in the North-South connection (NSC) in Brussels? How to make the timetable for the NSC more robust?

The busy traffic through the NSC impacts the performance of the whole railway system negatively. That is why it is important to focus on the robustness of the timetable for the Brussels' area. If the system becomes more robust in the Brussels' area, trains will leave this area with less delays than before, and thus fewer conflicts will arise on the remainder of their itinerary. As a consequence, the overall performance of all trains in the entire network should improve. The developed methodology to improve the robustness in the area of Brussels is presented in Chapters 4-7.

> **Research Question 3:** What is the effect on the robustness of a number of structural measures for the North-South connection (NSC)?

Nearly all trains that run through Brussels dwell at one of the six platforms of the Central station. As a consequence, this station seems the real, physical bottleneck of the entire area. Extending the capacity by creating more platforms will somewhat relax the throughput at the Central station. In Chapter 8, a "*what if*"-study is made in order to estimate the impact on the robustness in the NSC of this and other measures to decrease the capacity usage or to increase the available capacity through the NSC.

> **Research Question 4:** Can the developed approach be used for other bottlenecks in the network or for other timetable-related problems?

To validate the general applicability of the presented methodology, the developed algorithm is applied to some new case studies. The first one corresponds to an extension of the NSC network to include its surroundings. The second case study is based on the station area of Antwerp. This area is another bottleneck in the Belgian railway network. The fact that Antwerp Central station is a terminal for most trains makes this case study significantly different from that of Brussels. The obtained results are summarized in Chapter 8.

In Chapter 9, it is shown that a similar algorithm as the one of Chapters 4-7 can be used to reschedule the timetable in case of a temporary infrastructure unavailability. Doing so, the generality of the introduced approach to other problems is illustrated.

1.4 Approach and definitions

When a delayed train approaches a station, dispatchers have different options to limit the hinder caused by that train. Examples of possible actions are the rerouting of a train, decisions about the order of events like the departure of trains, and directing a train to another platform. Although this research focuses on the planning phase and not on the operational level where the dispatching happens, similar actions to improve the robustness in station areas can be considered during the planning phase.

In general, the construction of a railway timetable is done top-down. The schedule is created on a macroscopic level and microscopic checks are used to evaluate and repair the feasibility at individual stations (Huisman et al. 2005). The other way around, first building a plan for each station individually and combining all these (small) plans to one (large) timetable, the bottom-up approach, can be very cumbersome and non-efficient. Something in between this top-down and bottom-up approach can be practical especially in the case of star shaped networks. Using the nomenclature of the theory of constraints (drum-buffer-rope) (Goldratt 1984), one may say that the center functions as drum and the lines towards the center can act as buffer. Adapting the other stations in the network to the new planning (rope) and hitting the drum at the optimal beat of the center, can improve the performance of the overall system. This is the idea behind our study.

In order to construct a robust timetable, we propose to develop a timetable for the center of the network or for the main bottleneck. Next, this local timetable can be used as starting point to construct a timetable for the whole Belgian network. To extend the local timetable to a feasible global schedule, one can use a bottom-up approach with feedback loops or the technique of goal programming (Vansteenwegen et al. 2006, 2007). Another possibility is similar to the Swiss approach of scheduling in compensation zones and condensation zones. The compensation zones are the buffers between condensation zones which are bottlenecks like station areas. More background about this approach can be found in Caimi et al. (2009a) or Section 2.3.9.

The developed algorithm starts from an initially feasible schedule. Although changing the event times of some trains in a station may lead to conflicts at other stations, this is not sorted out in this research. In line with the arguments of Cacchiani et al. (2008b) for scheduling on corridors instead of an entire network, we are convinced about giving precedence to a good schedule for the central bottleneck and assume that these conflicts are not impossible to solve. For example, by using the spare capacity outside the considered station area. Moreover, in the general approach, timetables are created macroscopically and microscopic feasibility checks at station level are made afterwards. Nevertheless, the fact that conflicts can arise outside the considered area should be kept in mind and interpreting and extrapolating the obtained results should be done carefully.

Definitions

In this dissertation, the term **station area** is used to indicate the network that consists of a set of stations with their platforms and the tracks and switches that connect the incoming lines with the platforms and the outgoing lines. An overview of the station area of Brussels is given in Figure 1.1.

The **(minimum) time span** between two trains is defined as the smallest delay that causes these trains to conflict. By comparing the blocking times of the trains' common sections under undisturbed circumstances, the minimum time span between two trains can be obtained. When the minimum time span is nonnegative for all pairs of trains, the train schedule is considered **feasible** or **conflict-free**. In case of disturbed circumstances, conflicts between trains occur. The delays that originate due to these conflicts are called **knock-on delays**, **propagated delays**, or even **secondary delays**. The term **primary delays** is used for the delays that come from external causes like, for example, signal failures, overrun dwell times due to a large number of passengers, or bad weather conditions. Delays that are due to the uncertainty of the travel times are also primary delays.

The time needed to make a certain journey according to the timetable, so under ideal circumstances, is what is called the **planned travel time**. In reality, disturbances and conflicts will occur such that the time spent while traveling will be different from the planned travel time. Therefore, we define the **real travel time** as the travel time that is needed to make the trip under these disturbed circumstances, so from the moment of planned departure until the effective arrival at the destination. Next to the planned travel time and the real travel time, there is the **nominal travel time (NTT)**. The NTT is the time needed to make a journey when the duration of each action (including transfers) equals the minimum necessary duration. The difference between the NTT and the planned travel time is due to the inclusion of slack time (time

Figure 1.1: A schematic overview of the station area of Brussels[3]. The NSC is situated in the middle of the figure. The tracks to the top right are in the direction of Antwerp.

reserves) in the form of **supplements**, **buffers**, and scheduled waiting time that is needed for feasibility. Supplements are scheduled between two events of one and the same train. Buffers correspond to idle times between two events of two different trains. Based on this definition, the minimum time span between two trains actually corresponds to the smallest buffer between all events of these two trains. The use of supplements and buffers will be discussed thoroughly in the next chapter.

[3]Source: http://www.infrabel.be/sites/default/files/documents/ns_c-01-map-net-10459-01_1.pdf, consulted in September 2014

1.5 Main contributions and outline

In this section, an overview of the main contributions of this dissertation are given per chapter.

Review of the existing literature about railway planning and optimization.
Chapter 2 provides an overview of the available literature about railway planning and the different optimization problems that are faced in this research. First, the different levels of railway planning are introduced and then the methods and objectives that are related to this dissertation are described. Like for this research, the main focus of the literature study lies on the tactical level planning of the timetable and the routing.

Formulation of an all-embracing and practically usable definition of railway robustness.
Chapter 3 starts with collecting and discussing the robustness definitions that are found in literature. It is shown that these definitions correspond to various points of view which are not always compatible. Moreover, some drawbacks are identified. To bridge the gap between the different interpretations of robustness, an all-embracing definition is presented. This definition accounts for the passengers who are the clients of the passenger railway system, and is based on the passengers' valuation of travel time. The usability of this definition is illustrated by measuring the robustness of the 2010 timetable for the Belgian railway network.

Development of a methodology to improve the robustness in railway bottlenecks based on:

> **An objective function to spread the usage of resources in time.**
> *Chapter 4* introduces the framework of the algorithm that is used to improve the robustness. Since complicated and time consuming delay propagation computations are required to measure the robustness, a substitute objective function to guide the optimization phase is introduced. After presenting the details of the North-South connection (NSC) case study, an overview is given of the assumptions that are made.
>
> **The optimization of the routing of trains through the network.**
> *Chapter 5* explains the applied solution technique for solving the *train routing problem (TRP)*. The TRP is the problem of finding exactly one route for each train through the considered network in a conflict-free way. Using efficient preprocessing techniques, the optimal route for each train could be computed by formulating the TRP as a mixed integer linear problem (MILP). The term *routing module* is used to refer to this solution

approach. Chapter 5 ends by discussing the impact of the routing module on the robustness of the NSC case study and by comparing the developed methodology with routing techniques from the literature.

An integrated approach to exploit the potential of changing train orders and schedules.
Chapter 6 is about the optimization of the timetable, the so-called *train timetabling problem (TTP)* or, in this dissertation, the *timetabling module*. A tabu search algorithm is developed that considers the shift of one or more trains in time and the swapping of the order of two trains. Special emphasis is given to the fact that an order swap incurs a time shift for (at least) one train which can enable the possibility to improve the schedule for other trains. At the end of this chapter, computational results show that the robustness of the timetable can be increased considerably by using the developed tabu search algorithm.

The potential of modifying the platform allocations.
In *Chapter 7*, the selected platform for each train in each station is questioned. By allocating a train to a different platform, one can anticipate on conflicts that are likely to occur during operation. As a consequence, a new route from and to the new platform is required and different infrastructure resources are used. To resolve conflicts, for example, if a part of the new route is already reserved for another train, or to test the *potential* of this platform reallocation for other trains, the timetabling module from Chapter 6 is repeated before evaluating the entire platform change. In the end, simulation results prove the effectiveness of the *platforming module* and the entire algorithm for the NSC case study.

A significant decrease of the delay propagation for different case studies.
Chapter 8 contains the computational results of the entire algorithm for a number of case studies. The influence of measures that alter the available capacity or the capacity usage is assessed for the case study of the NSC. Among others, an increased number of platforms in the Central station or a new stopping pattern in that station are considered.
In the second part of chapter 8, the network of the station area of Brussels is significantly extended, and the impact of the developed algorithm on the case study of Antwerp is discussed. In comparison with the previous case study, more details are included in the latter two and the results are closer to reality. This is confirmed by a validation study using one of Infrabel's commercial simulation packages that endorsed our findings.

Illustration of the general applicability of the developed methodology.
Chapter 9 introduces the problem of replanning the train timetable to adapt for a planned infrastructure unavailability. If certain tracks in the network become unaccessible due to planned maintenance actions, the schedule and routing of some trains needs to be modified to avoid these tracks. Moreover, since the capacity is reduced, the system becomes more vulnerable to delay propagation and thus less robust. In this chapter, it is shown that the presented interpretation of robustness and the developed algorithm can be adapted to cope with this kind of problems.

Chapter 10 concludes this dissertation with a summary of the obtained results, an answer to the research questions, and some ideas for further research.

Chapter 2

Literature survey

In science, read by preference the newest works. In literature, read the oldest. The classics are always modern.

Amy Lowell

The work done for this dissertation is preceded by several studies in the literature. This chapter provides an overview of the related articles about these studies. Since the next chapter is devoted to the concept of robustness, the discussion about robustness definitions in railway optimization can be found there. The same holds for the literature about the impact of maintenance on the railway system, which is grouped in Chapter 9.

2.1 Introduction on railway planning

Based on the planning horizon, various problems turn up when planning a railway system. In Figure 2.1, a graphical overview of the different steps in railway planning is given. This figure comes from the paper of Lusby et al. (2011a). This paper, as well as the ones of Caprara et al. (2007) and Huisman et al. (2005), provides an extensive overview of all planning problems in passenger railways. In this section, the main components are introduced. The above mentioned papers contain further references about the specific problems and Cordeau et al. (1998) sketch more background about the scheduling of freight trains.

This chapter partially corresponds to Dewilde et al. (2011).

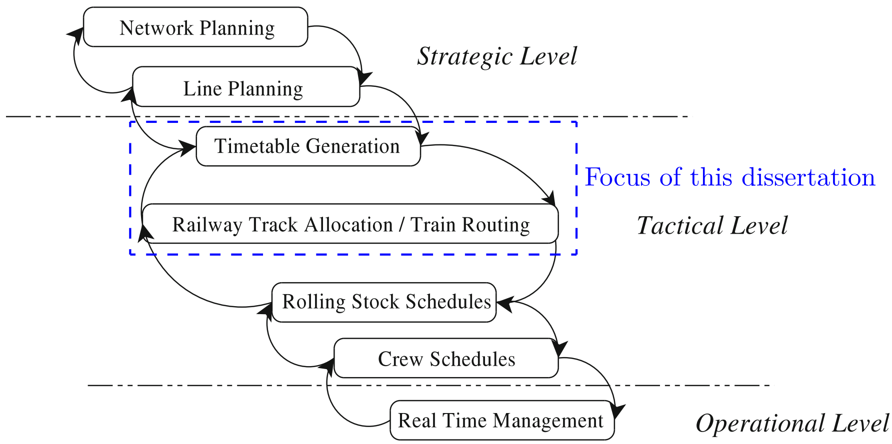


Figure 2.1: The different steps in railway planning according to Lusby et al. (2011a). The planning steps handled in this dissertation are highlighted.

On the long term, in the strategic level planning phase, fundamental decisions are made. These decisions concern the infrastructure and the way stations are connected: with a direct connection or by means of one or more transfers. In the *network planning step*, the location of stations, the number of platforms and their length, and the number of parallel tracks between two stations are determined. Also infrastructure changes such as the creation of a non-level crossing by building a bridge are assessed in this step. More references about the network planning step can be found in Engelhardt-Funke et al. (2004), Mesa et al. (2013), and Schöbel (2005).

When the railway network is known, the *line planning* can be solved. Huisman et al. (2005) define a *line* as *a direct railway connection between two end stations that is operated with a certain frequency and with a certain train type.* This means that, when creating the line planning, one decides from where to where trains ride, at what stations they stop, how often this happens, and according to which regime these trains ride. Typically, the options for the latter are: international trains, intercity trains, interregional trains, and local commuter trains. In some countries, extra train types like, for example, peak hour trains exist. The line planning has been studied by, among others, Bessas et al. (2009), Bussieck (1998), and Goossens et al. (2006).

The tactical level or medium term planning phase ranges from a couple of days to several months or even a year. During this phase, one schedules the usage of the available resources to fulfill the line planning. The first two problems that are considered at this level are the *timetable generation* or *train timetabling problem (TTP)* and the *railway track allocation* and/or *train routing problem (TRP)*. Both are discussed more thoroughly later in this chapter. The other two problems that are solved, concern the scheduling of the *rolling stock* (see for example in Cacchiani et al. 2008a; Cadarso et al. 2011; Kroon et al. 2008b) and the *crew* (like in Bengtsson et al. 2007; Folkmann et al. 2007). The former is the problem of assigning one physical train with the right composition, for example, one locomotive and a number of carriages, to each planned trip. The latter is about selecting the necessary crew members (driver and guards) to operate the train.

When the schedule is effectively operated, *real-time management* is needed to solve conflicts and to ensure that the safety regulations are respected. Real-time rescheduling is the subject of the work of, among others, Cacchiani et al. (2014), Caimi et al. (2012), Corman (2010), and D'Ariano (2008). When delays influence the transfer reliability, one comes in the field of delay management which is about the waiting decision for transfers (see Biederbick et al. 2007; Dollevoet et al. 2012; Meng et al. 2011, to name some). Operational level problems also deal with conflicts in the rolling stock or crew schedules like studied by, for example, Huisman (2007), Maróti (2006), Nielsen et al. (2012), and Veelenturf et al. (2012).

The downward arrows in Figure 2.1 indicate the general order in which the problems are solved. Often, a backtracking step or some iterations are required to ensure that a feasible or better solution can be found for a problem downstream. There are some approaches that integrate two planning steps. For example, in Marín et al. (2009), the network planning and line planning are considered simultaneously; the interaction between the line planning and the TTP is studied by Goerigk et al. (2012) and Kaspi et al. (2013); and solving the TTP together with the rolling stock assignment is done by Cadarso et al. (2012, 2013).

2.2 Objectives for railway optimization

The TTP is not only about determining arrival and departure times for each train in each station such that a conflict-free schedule is obtained, but also about finding the best timetable with respect to one or more objectives. On the one hand, from the point of view of the railway companies (the suppliers), there are cost-related objectives such as minimizing the operational cost (Lindner et al. 2005). On the other hand, from the travelers' perspective, there are objectives that consider performance criteria. A natural one is to minimize the average delays (Kroon et al. 2008d). Other possibilities vary from minimizing the waiting time (Liebchen 2008; Nachtigall 1996; Wong et al. 2008), over minimizing the percentage of missed transfers (Liebchen et al. 2010), to minimizing the time until delays are absorbed (Goverde 2007). Only few authors explicitly account for passengers in their objective function. Possible passenger oriented objective functions are to minimize the total delays of all passengers (Kroon et al. 2007a), to optimize transfers (Schöbel et al. 2009; Vansteenwegen et al. 2006, 2007), to minimize the planned waiting time (Engelhardt-Funke et al. 2004), or to minimize the planned travel time of all the passengers in the system (Goerigk et al. 2010; Schmidt et al. 2014). The latter is often used without considering the passengers explicitly but by adding weights based on, for example, the train and the location, to approximate passenger numbers relatively (Ghoseiri et al. 2004; Liu et al. 2009; Zhou et al. 2007).

Optimizing the robustness of a system is another popular objective when designing a railway timetable. However, many different implicit and explicit definitions of robustness have been presented. These definitions and their implications will be discussed in detail in the next chapter.

2.3 Solution approaches for the train timetabling problem

In most countries with a dense railway network, timetables are *cyclic* and *symmetric*. The same holds for Belgium. Cyclic or periodic means that the schedule repeats itself after a certain period, for example, one hour. This makes it easier for the passengers to remember their departure and arrival times. In Liebchen (2004), it is said that a timetable is symmetric whenever two trains of the same series with opposite orientations always meet at an integer multiple of the period. Graphically, this means that the trajectories of these trains form each other's mirror image in time-distance diagrams. Symmetry can only be imposed if the running and dwell times in both directions of a line are equal. A consequence of symmetry is that in each station, the sum of the arrival time of a train in one direction and the departure time of the train in the reverse direction equals an integer multiple of the period, and that the travel times and the transfer times are the same in both directions. This also makes the chance of missing a connection smaller than in the case of asymmetric transfer times.

Railways have a large history and the timetabling process has changed a lot over time. Nowadays, there are several ways to construct a timetable. One of the main difficulties is that the real duration of a ride or dwell action is not known in advance. In literature about railway scheduling, several methods of how to deal with this inconvenience have been presented (Cacchiani et al.

2012b). In this section, the most important ones are presented. The differences in methods mainly come down to how extra scheduled (slack) times are inserted in the timetable.

2.3.1 Supplements and buffers

In general, three different types of extra scheduled times (timetable slack) are being used. Next to the *scheduled waiting time* that is needed for feasibility, there are the *supplements* and *buffers* (Kroon et al. 2008a). In this dissertation, the terms *slack time* and *time reserves* are sometimes used instead of supplements and buffers. Supplements are extra planned times between the arrival and departure events of the same train, whereas buffers are extra planned times between two arrival or departure events of two different trains.

A supplement can be implemented on a trip between two stations (*running time supplement (RTS)*) or during a dwell action (*dwell time supplement*) to anticipate a longer alighting and boarding time than normal. The inclusion of an RTS prolongs the planned travel time of the passengers but enlarges the possibility to arrive on time. A similar argument applies for the dwell time supplements and the following departure event.

In contrast with a supplement, a buffer does not necessary prolong the itinerary of a passenger. For example, a *headway buffer* ensures more spacing between two consecutive trains on a common part of the infrastructure and does not need to affect the planned travel time. Another kind of buffer is the *transfer buffer* which is a surplus on top of the minimum transfer time. The inclusion of a transfer buffer decreases the chance that a transfer will be missed at the cost of a longer (planned) travel time for the transferring passengers.

Almost all railway timetables contain extra scheduled times. Nevertheless, their total amount and the procedure to distribute the time reserves vary a lot.

2.3.2 The period event scheduling problem

A first method to compute a railway timetable is by solving the periodic event scheduling problem (PESP) that is introduced by Serafini et al. (1989). As the name reveals, in the periodic event scheduling problem, departure and arrival events have to be scheduled in a cyclic way. Denote the period with H, an event with i or j, and the time assigned to event i with π_i for which $0 \leq \pi_i < H$ holds. Let l_{ij} and u_{ij} be the lower and upper bound for the time between events i and j, respectively. Using this notation, it is possible to model the PESP constraints as follows

$$l_{ij} \leq \pi_j - \pi_i + k_{ij} \cdot H \leq u_{ij}, \tag{2.1}$$

for all related events i and j, and a $k_{ij} \in \mathbb{Z}$ or $k_{ij} \in \{0,1\}$ if $[l_{ij}, u_{ij}] \subset [0, H)$. The integers k_{ij} are called phase shifts and are necessary to determine the cyclic order of events. The usage of these integer variables makes the PESP hard to solve. In Vansteenwegen (2008), the variables k_{ij} are omitted by considering multiple consecutive periods. A second approach to avoid these variables is by using the process times instead of the event times. A process time or periodic tension (x) is defined as the difference between event times

$$x_{i,j} = \pi_j - \pi_i + k_{ij} \cdot H.$$

The usage of this tension resulted in the cycle periodicity formulation (CPF) that is studied by, among others, Liebchen et al. (2008) and Nachtigall (1996). Compared to the standard PESP formulation, the CPF has less integer variables and an extra constraint can be added stating that the sum of the process times along any directed cycle is an integer multiple of the period length. As a consequence, the CPF has a better linear programming (LP) relaxation and is easier to solve.

Initially, the PESP is designed for finding a feasible timetable. Since the PESP does not account for delays, the most frequently used objective is to minimize the (passengers') planned travel time (Goerigk et al. 2010; Liebchen et al. 2009; Nachtigall et al. 1997). This can be formulated as

$$\sum_{i,j} w_{ij}(\pi_j - \pi_i), \tag{2.2}$$

with w_{ij} the number of passengers that start their journey with event i and end it with event j. Probably the best known application of the PESP is the Dutch CADANS system to generate railway timetables that is developed by Schrijver et al. (1994). In Hooghiemstra et al. (1999) and Kroon et al. (2009), more information is given about the implementation and success of DONS, the *Designer Of Network Schedules* tool CADANS is part of. For more details about the PESP, the CPF, and possible extensions, the reader is referred to Cacchiani et al. (2012b), Liebchen et al. (2007), and L. Peeters (2003).

2.3.3 Nominal timetable

Using the minimum necessary process times turns constraint (2.1) into an equality constraint for each ride, dwell, or transfer action; for each (i, j) that represents such an action, l_{ij} becomes equal to u_{ij} in (2.1). In order to maintain feasibility, scheduled waiting time implying that $l_{ij} < u_{ij}$ can be necessary in some cases. Planning the ideal process times gives the fastest schedule in theory, however, this is very bad since the smallest delays will cause a knock-on effect.

2.3.4 Robust optimization

The next possibility is the opposite of the nominal solution; a schedule that remains conflict-free, even in the worst-case scenario. This approach is called robust optimization and is considered by, for example, Ben-Tal et al. (1998) and Bertsimas et al. (2004). In order to anticipate all worst-case scenarios, a lot of time reserves are needed. Although timetables that are constructed this way are very stable, they are not attractive since they are mostly much too slow. This is an important remark: although a timetable is insensitive to delays, it can still be very unattractive.

2.3.5 Light robustness

A solution of the TTP that lays somewhere in between the nominal solution and the timetable that results from robust optimization is possible. The idea of light robustness (Fischetti et al. 2009a,b) is to allow a worsening in objective function value compared to the nominal timetable and to use this worsening to improve the timetable by, for example, the stochastic programming approach of Section 2.3.7.

Let z^* be the optimal objective value of the nominal solution with objective function (2.2), then the upper bound for the planned travel time of the light robustness solution is modeled as

$$\sum_{i,j} w_{ij}(\pi_j - \pi_i) \leq (1 + \delta)z^*. \tag{2.3}$$

Notice that for $\delta = 0$, the nominal solution is optimal, and that for $\delta = \infty$, this approach is equivalent to robust optimization. By adding supplements, the timetable becomes slower and the left-hand side of (2.3) increases. This is the price for fewer conflicts.

2.3.6 Recoverable robustness

The method of recoverable robustness is introduced by Cicerone et al. (2009) and Liebchen et al. (2009). It is initially developed to deal with large delays, but the approach can even be used for general scheduling purposes. The idea comes from the observation that, when constructing a timetable, schedulers do not account for repair strategies like how to restore feasibility in case a train breaks down in the middle of a bottleneck. The method of recoverable robustness works as follows. In the TTP, a timetable and a recovery algorithm to solve conflicts are determined such that, when operating the schedule, the recovery algorithm can be used to regain feasibility as soon as possible. The goal is to find a timetable and a recovery algorithm such that a conflict-free timetable can be obtained in all considered scenarios.

Recoverable robustness can be applied to various railway planning problems. For example, it is applied to solve the train platforming problem (TPP), see below, by Liebchen et al. (2009) and for scheduling the rolling stock like in Cacchiani et al. (2008a, 2014). Another version of recoverable robustness for the TTP, called recover to optimality, is presented by Goerigk et al. (2010). The idea of recover to optimality is to find the timetable for which the recovery costs are as small as possible. Just like recoverable robustness, the dynamic traffic management system of Schaafsma (2001) and Schaafsma et al. (2007) leaves certain decisions for the operational phase. In Caimi et al. (2011b) and D'Ariano et al. (2008b), flexibility is inserted in the timetable, for example, to postpone ordering decisions to the point in time where more information from the operations becomes available.

2.3.7 Stochastic programming

Another approach to schedule the arrival and departure activities of all trains in a network is stochastic programming (Kroon et al. 2007a, 2008d; Vromans 2005). Stochastic programming makes use of a distribution of delays and returns the timetable that gives the best results on average, for example, the timetable with the lowest delays on average. In general, there are two types of stochastic programming: chance constrained and multi-stage programming with recourse. In chance constraint programming the solution that is conflict-free in as many scenarios as possible is searched, whereas in each stage of multi-stage programming, a solution is evaluated and adapted using information that was not known before.

The most used approach for the TTP is two-stage programming with recourse. In the first stage, a timetable is constructed, and in the second stage, delays are added to the model and used to evaluate the solution of the previous stage. Doing so, there is a kind of simulation process built into the model. As input data, a set of primary delays is used. In Kroon et al. (2008d), it is shown that the results are rather robust against small changes in input data. The main drawback of this method is its complexity that causes very large computation times. This is due to delay propagation computations that are required to evaluate the objective function.

The technique of stochastic programming is used to find the optimal allocation of supplements for a case study in the Netherlands (Kroon et al. 2007a, 2008d; Vromans 2005). Starting from an input timetable, the model searches for the reallocation of the available supplements to minimize the average delays. To avoid the usage of phase shift variables like in the PESP, the cyclic order of events could not be changed during this reallocation process. In Kroon et al. (2008d), some results are presented for a small part of the network. The average punctuality, on a 3 minutes base, increased from 86.7% to about 93%.

2.3.8 Goal programming

Goal programming is used by Vansteenwegen (2008) and Vansteenwegen et al. (2006, 2007) to construct timetables with an improved transfer schedule. Instead of adding transfer buffers, each transfer is optimized individually by assessing the pros and cons of adding an RTS on the trip leading to the transfer station. Afterwards, a schedule for the whole network is constructed for which the planned travel times are as close as possible to the optimal ones determined in the previous step. Using goal programming, partial schedules can easily be combined in a timetable for the whole network.

Providing an RTS on the track section leading to the transfer station increases the punctuality of the feeder train. Thus, the transferring passengers benefit since the available time to make a transfer grows and the probability of a successful transfer increases. For the passengers who reach their destination, their actual arriving time does not change, but their feeling of punctuality changes; since the planned arrival time is now postponed with the supplement, they experience less delays and are more satisfied. For through passengers this is different. They may experience disadvantages since their train may dwell longer than necessary. The same holds for departing passengers in the opposite direction. Due to symmetry, the RTS in one direction turns into a maximum holding time for the same train in the opposite direction which is then the connector. Notice that this implies that only one supplement per transfer is needed. Computational experiments on a real network resulted in a reduction in waiting cost by 40%.

2.3.9 Scheduling with compensation and condensation zones

To schedule an entire network, Caimi et al. (2009a) came up with the idea to divide the network in compensation zones and condensation zones and to schedule each of these zones individually. Typical examples of condensation zones are station areas with dense railway traffic. Compensation zones are the less busy parts of the network that connect the condensation zones. Due to the scarce capacity, the complex infrastructure, and the large amount of trains, there is much interaction between trains in condensation zones. Therefore, the routing of the trains through these zones is important. Since this is the topic of Section 2.4, the description of scheduling in condensation zones is added there. In compensation zones, however, traffic is less dense and the interaction between trains mainly comes from predictable actions like an overtaking or a single track conflict. According to Caimi et al. (2009b), the timing and speed of each train at the borders of each zone are predetermined such that each subproblem can be solved individually. As objective for the scheduling in compensation zones, they propose to minimize the energy consumption or to improve the reliability of the timetable.

2.3.10 Job shop scheduling

A last method that is presented, is based on the formulation of the TTP as a job shop scheduling problem. The first to point at the similarities between both problems was Szpigel (1973). By using the job shop formulation, the TTP can be modeled using a disjunctive graph. For the tactical level TTP, Khosravi et al. (2012) developed a solution technique based on a modified shifting bottleneck procedure. Their heuristic is tested on a dense and complex network with two terminal stations with predetermined routes, and the performance is compared with a deadlock-prone first-come, first-serve approach. As objective, they minimize the total weighted tardiness of each job (train). The long computation times make this approach not suited for the operational level. This is in contrast with the alternative graph formulation of Mascis et al. (2002) that served as input for the real-time rescheduling (and rerouting) using the job shop representation by Corman et al. (2011, 2009), D'Ariano et al. (2008a, 2007), and Mazzarello et al. (2007).

2.3.11 Conclusion

Most of the existing timetabling approaches that are discussed in this section aim at designing a completely new schedule and approach the problem from a macroscopic point of view, while we include detailed infrastructure information with capacity constraints and start from an already existing timetable. Therefore, none of the above presented methods is used in this dissertation, but the ideas behind these methods and the encountered difficulties such as the tough ordering decisions are a good source of inspiration for the development of our own timetabling method (Chapter 6).

2.4 Train routing and platforming problem

According to Figure 2.1, solving the train routing problem (TRP) follows after the TTP. The TRP distinguishes itself from the train platforming problem (TPP), which is the term that is used in case the choice of platform uniquely determines the route through the station area. In this dissertation, the focus is on large and complex station areas where multiple routes exist to travel from one end to another end, thus the term train routing is more appropriate. Where in the TTP the infrastructure is regarded from a macroscopic point of view, the track layout is considered at microscopic level in the TRP, so much more details are taken into account.

The TRP is studied by several authors and differences exist in the considered level of planning, the approach, the choice of objective function, the option to allow small timetable changes, etc. An extensive review paper, written by Lusby et al. (2011a), categorizes the related literature based on the approach. In this section, the overview is restricted to articles that are the most relevant for this dissertation.

2.4.1 The node packing approach of Zwaneveld

Some of the first authors that modeled the TRP are Kroon et al. (1997), Zwaneveld (1997), and Zwaneveld et al. (2001, 1996). In the remainder of this dissertation, we will refer to this approach as the *approach of Zwaneveld*. Where Schrijver et al. (1994) use the PESP to design the CADANS module of DONS, the authors behind the approach of Zwaneveld use a node packing formulation based on a conflict graph to solve the TRP in the STATIONS module of DONS. Initially, the STATIONS module is developed as a strategic tool to analyze future capacity requirements. At that time, the problem was seen as a feasibility problem which is NP-complete (Kroon et al. 1997; Zwaneveld et al. 1996). To increase the probability of finding a feasible solution, timetable deviations up to one minute are considered. This increases the size of the model since one variable is needed for each combination of a train with a candidate route and a specific time. Later on, the tactical level variant of routing trains gained in interest. In the follow-up paper, Zwaneveld et al. (2001) add weights to each train and strive to maximize the number of routed trains, followed by the minimization of the number of shunt movements and deviations from the preferred platform.

The resulting problem can be modeled as a conflict graph and solved as a node packing problem (NPP). In a conflict graph each variable is represented by a node and all pairs of conflicting variables are connected by a direct edge. The NPP is then about finding the largest set of non-adjacent nodes. Since the NPP is known for its weak LP relaxation, Zwaneveld et al. (2001) add clique inequalities to the model and perform some preprocessing to reduce the problem size. Nodes that are dominated by other nodes (node dominance) or a set of other nodes (set dominance and iterative set dominance) are removed from the problem instance without affecting the quality of the final solution.

2.4.2 Extensions of the approach of Zwaneveld

The approach based on the NPP is extended to include robustness by Caimi et al. (2005, 2009a, 2011a). The high interaction rate in the condensation zones, see Section 2.3.9, requires integration of timetabling and routing. To include the timetabling part, timetable deviations are required. Inserting a time-index and accounting for a shift of the most critical point between two trains, the resulting model became too large to be solved by exact techniques. Therefore, a fixed-point iteration heuristic is developed. A robustness oriented objective function is added to maximize the four smallest minimum time spans between any two trains. A similar approach is used by Burkolter (2005) and Herrmann (2006).

In Kroon et al. (2008c), another technique to obtain robust train routes is considered. The authors start from the model of Zwaneveld et al. (1996) but include a robustness oriented objective function. In order to keep their objective function linear, they investigate several linearisation techniques. Based on the size of the resulting model and the quality of the LP relaxation, they suggest a combined linearisation technique. However, due to the large number of variables and constraints, the level of detail, and the number of possible routes, they are unable to solve the entire model for real-life instances using a standard mixed integer problem (MIP) solver. Therefore, it is decided to aggregate parts of the infrastructure and to solve the TRP on the aggregated level.

2.4.3 Other routing approaches

Other approaches to solve the TRP are by formulating it as a set-packing model as presented by Lusby et al. (2011b), using constraint programming as is done by Delorme et al. (2001) and Rodriguez (2007), or by heuristical approaches (see Carey et al. 2000, 2007). The operational level TRP can be solved using the alternative graph formulation of Corman et al. (2010) and D'Ariano et al. (2008a).

The network that is considered in Rodriguez (2007) is concentrated around a terminal station. In such a station, trains arrive and depart over the same tracks and spend more time at the platform such that the capacity consumption is much larger than in through stations. The case study used in Carey et al. (2000, 2007) consists of a network of stations that are lined up. It is assumed, however, that the routes are determined by the chosen platforms. Thus, this is more a TPP approach. The TPP, which is actually a special case of the TRP, is optimized to minimize costs by Caprara et al. (2011), with respect to track usage by Billionnet (2003), or for the purpose of capacity studies by Cornelsen et al. (2007) and Sels et al. (2014, 2011c). In Sels et al. (2014), an MILP model is developed to platform as many trains as possible from a set of current and potential, future trains. To evaluate the station capacity in practice, the model of Billionnet (2003) is extended to handle differences in route durations and train lengths. Another feature is that split and merging actions are considered too. The objective function penalizes the usage of a dummy platform (with unlimited capacity) and, to a smaller extent, deviations from a preferred platform. The method's applicability is proven by test cases for a number of real stations.

2.4.4 Conclusion

Because of the focus of this dissertation on large and complex station areas, determining optimal routes through these station areas is an important aspect within the tactical level planning. Therefore, the approach of Zwaneveld and the robustness extensions of Caimi et al. (2005) and Kroon et al. (2008c) will serve as input for the routing model that is developed in Chapter 5.

2.5 Evaluation using simulation

There are several ways to evaluate the outcome of the planning process. Some authors use analytical measures for a quick evaluation of some properties. However, not all performance measures can be computed easily. A typical example is delay propagation, which is hard to compute. Therefore, simulation models can be used. In this section, some background about simulation and its possibilities is sketched. Since simulation will not be a part of our solution approach, but just a tool used for evaluation, no details are discussed here.

Simulation models have several characteristics. The most important ones are the analytical approach (deterministic or stochastic), the time steps (discrete or continuous), the processing (synchronous or asynchronous), and the scale (macroscopic or microscopic). A description of each of these characteristics is given by Siefer (2008). Simulation tools can be used during all planning phases. On the strategic level, one can make capacity evaluations and cost-benefit analyses of infrastructure investments (Delorme et al. 2009; Engelhardt-Funke et al. 2004). The performance of the timetable and routing that are generated at the tactical level can be evaluated such as in Carey et al. (2000), Franke et al. (2013), and Middelkoop et al. (2001). The impact of dispatching strategies can be studied by means of operational level simulation (Lee 1998; Middelkoop et al. 2006; Salido et al. 2012). Next to simulating the train operations, the travel behavior of the passengers has been imitated by Kanai et al. (2011) and Kunimatsu et al. (2012). The main advantage of simulation is that it is much cheaper than real tests. Moreover, not all experiments can be done in practice, while all kind of scenarios can be provoked over and over again with simulation. Even for the purpose of training, feasibility checks, or for the comparison of strategies, simulation is suitable.

A lot of commercial software to simulate the performance of railway systems exists. An overview is given by Barber et al. (2007). Next to the technique of Monte Carlo simulation based on discrete event systems (Nash et al. 2004; Siefer et al. 2005; Takeuchi et al. 2007), several other methods are used to build simulation tools like queueing models (Engelhardt-Funke et al. 2004; Janecek et al. 2010) and stochastic models (Carey et al. 2000; Franke et al. 2013). Also analytical measures served as a basis for simulation, for example, the theory of max-plus algebra (Goverde 2002, 2007) or the stochastic modeling of delay propagation of Büker et al. (2012).

2.6 Summary

The entire railway planning process consists of several steps. In general, the problems corresponding to each step are solved one by one, and sometimes, a backtracking step is required to obtain overall feasible or better solutions. This dissertation deals with two problems of the entire planning process: the train timetabling problem (TTP) and the train routing problem (TRP). When solving these problems, one mainly tries to minimize delays, to minimize the scheduled waiting times, or to minimize the planned travel times. After introducing the notion of supplements and buffers, an overview of several methods to develop or improve a timetable is given. The methods that are worked out in this dissertation are, at first, intended for large and complex railway stations. Therefore, determining the optimal routes through these station areas is an important aspect within the tactical level planning. The most relevant approaches that are used to solve the TRP are discussed. A brief overview of simulation techniques concludes this chapter.

This literature review showed that including a full evaluation of performance indicators like knock-on delays in the optimization algorithm would result in a cumbersome and time-consuming method. As a consequence, a simplified objective function, in combination with an independent simulation model can be more appropriate. The former allows quick and easy evaluations and provides a good guiding of the algorithm, while the latter can be used to assess the quality of the system in detail after the optimization. Since our algorithm starts from an existing timetable and approach the problem from a macroscopic point of view, not all presented techniques are found to be useful. Nevertheless, the modeling approach and the encountered difficulties such as the tough ordering decisions are a good source of inspiration for the development of a module to improve the given timetable. When considering the routing of trains through a station area, the approach of Zwaneveld and the robustness extensions of Caimi et al. (2005) and Kroon et al. (2008c) will serve as input for our model.

Chapter 3

Comprehensive definition of robustness

> *A robustness definition has to be practical in the sense that railway companies are willing to use it. This means that striving for robustness should not make the timetable too slow, and a robustness definition has to be practical to work with such that the robustness can be assessed and compared. Thus, besides a definition of robustness, an appropriate way to measure and compare timetable robustness is needed.*
>
> Salido et al. (2012)

Based on a study about timetable evaluation criteria for railway operators and railway infrastructure managers, Schittenhelm (2011) concluded that robustness is the most prioritized criterion. But what is robustness? This chapter contains an overview of the different interpretations of railway robustness that are found in literature. Starting from quotations, different points of view are analyzed and some drawbacks are indicated. Based on these findings, a new and comprehensive definition of railway robustness is formulated. After illustrating the practical usability of this definition, some of the advantages and disadvantages of using the new definition are discussed.

This chapter is an updated version of the work presented in Dewilde et al. (2011, 2013).

3.1 Robustness definitions in literature

As indicated by Cacchiani et al. (2012b) in their survey on robust train timetabling, there is a large set of papers about timetable robustness in which each author captures the notion of robustness in his own way.

3.1.1 Delay propagation

One of the most important reasons why timetables can be classified as non-robust is delay propagation.

> *When a railway system is not robust, small external influences cause large delays which propagate quickly throughout the system in place and time. [...] Less delay propagation means a more robust timetable.* *(Vromans 2005)*

From this, one learns that avoiding the knock-on effect is a requirement for a timetable to be robust. As long as a train is delayed, it will infect the station areas on its route. As a consequence, the delays will spread out in space and time. By catching up its original schedule, the propagation of delays will fade. Thus one should take knock-on delays into account during the timetabling step. Given the total delays, it is, however, not easy to separate primary and secondary delays. Therefore, minimizing the propagation of delays is hardly used as objective function (Goverde 2005).

Many authors link robustness and delay propagation, for example, in the following quotation.

> *Timetabling robustness is not concerned with major disruptions (which have to be handled by the real time control system and require human intervention) but is a way to control delay propagation, i.e., a robust timetable has to favor delay compensation without human action.* *(Fischetti et al. 2009b)*

In this research, disturbances are interpreted as the daily occurring delays that are inherent to the system. Accidents, trains that break down, or other causes of large delays or interruptions of traffic are ignored since these cannot be accounted for when making the timetable.

In Salido et al. (2012), Vromans (2005), and Vromans et al. (2006), the heterogeneity of the timetable is analyzed. In Salido et al. (2012), analytical measures to evaluate the robustness on single-track lines with or without overtaking possibilities are proposed. Vromans (2005) and Vromans et al. (2006) homogenize the stopping pattern and the speed of the trains to achieve more freedom when scheduling. This concept is applied for one corridor at a time. This is in contrast with the analytical approach of Goverde (2005, 2007) who studies an entire network. The railway system is modeled as a discrete event system using timed event graphs (petri-nets) and max-plus algebra. Doing so, one can easily evaluate a system's sensitivity to delays and its stability, which is defined as the ability to recover from delays.

The major drawback of avoiding knock-on delays is that it solely focuses on delays and leaves passengers aside. This way, knocking on delays to a full train can be preferred above delaying an empty train.

3.1.2 Slack as tool for a robust timetable

Although simply focusing on the propagation of delays has some disadvantages, it is undeniable that knock-on delays play an important role in robustness. Formulating their vision, Kroon et al. (2008a) state that robustness is more than delay propagation.

> *Robustness of a timetable has one or more of the following effects (i) initial disturbances can be absorbed to some extent so that they do not lead to delays, (ii) there are few knock-on delays from one train to another, and (iii) delays disappear quickly, possibly with light dispatching measures.*
> *Both (i) and (iii) are a consequence of appropriately placed time supplements in the timetable, and (ii) is a consequence of appropriately placed buffer times between consecutive trains at certain locations. Note that, with light dispatching measures only, a timetable can only be robust against small disturbances.*
> *(Kroon et al. 2008a)*

The interesting thing in this definition is that, first, three potential properties of a robust timetable are given, and afterwards, it is described how these properties can be achieved. Since time supplements provide more time than needed for a ride or dwell action, they can be used to recover from disturbances or delays from previous actions. Buffer times increase the idle time between two trains and provide a better spacing such that the amount of conflicts decreases. Therefore, less propagation of delays takes place due to the usage of buffers.

Although three potential properties of a robust timetable are given, there is no answer to the question whether these three properties guarantee robustness. For example, the timetable that results from robust optimization satisfies all three criteria and can be called robust (in the strict sense of the word), but it is unlikely that the passengers and/or the railway companies are attracted to this (slow) schedule. Thus, it remains unclear what the right amount of slack is.

Another disadvantage of this definition is that there is no distinction between the delays of the feeder train of a popular transfer and the delays of the connecting train of that transfer, while the impact of the former is much larger if the transfer is broken. Thus, when considering robustness from a passengers' viewpoint, it is better to consider the average delays of the passengers instead of the delays of trains. Doing so, transfers play an important role.

3.1.3 Transfers as main component of robustness

By now it is clear that next to delays and knock-on delays, transfer reliability is important for the robustness of a system. An interpretation of robustness that focuses on transfers comes from Schöbel et al. (2009) who use a bi-criteria objective for the TTP: minimize the passengers' planned travel time and maximize the robustness.

> *A timetable has the robustness R if all its transfers are maintained whenever all source delays are smaller than or equal to R. [...] (Robustness measure) $R_{del}(V)$ is the maximal sum of all passengers' delay if all source delays are smaller than V.*
>
> *(Schöbel et al. 2009)*

Maximizing the robustness corresponds to maximizing R or minimizing $R_{del}(V)$. The delays due to a missed transfer are approximated by the period length. Since Schöbel et al. (2009) consider the passengers and their planned travel time as well as delays due to missed transfers, they turn robustness into *passenger robustness.*

Bi-criteria optimization is also applied by Cacchiani et al. (2008b) and Caprara et al. (2002) who use lagrangian optimization to combine their two objectives. The first objective aims at minimizing the deviations from an ideal schedule. The second objective arranges the allocation of slack times in the timetable.

3.1.4 Passenger service

Schöbel et al. (2009) evaluate robustness based on a fixed waiting time rule. Accounting for decisions about waiting for a transfer or not, one arrives at the field of delay management. Liebchen et al. (2010) aim to create a delay resistant timetable for which the difference in serviceability is as small as possible compared with the optimal delay management strategy for the considered situations. The research question they address is the following.

> *The question is whether such (delay resistant) timetables keep their promise, namely that disturbances do not affect the quality of service to the passengers too much if good delay management strategies are finally applied.* *(Liebchen et al. 2010)*

Although Liebchen et al. (2010), as well as Kanai et al. (2011) and Tomii et al. (2005) with their passengers' dissatisfaction, do not use the term robustness for it, they actually hit the nail on the head about what railway robustness is supposed to be. Since the most important goal of railway companies should be to serve passengers, one should, when talking about robustness, be talking about passenger robustness and what the passengers want. It is obvious that they do not want any delays and that they want to have good transfers. All of this can be captured in the idea of serviceability. Therefore, a possible definition of robustness can be

> *Robustness is the ability of a timetable to keep its level of service.* *(translated from Snelders et al. 2004)*

The main difficulty of working with this definition is the exact meaning of the concept of serviceability: What does it cover? How can it be measured?

In Liebchen et al. (2010), the service is improved by penalizing missed transfers whilst minimizing the passengers' planned travel time. Like in Vansteenwegen et al. (2006, 2007), transfers are optimized using passenger numbers and delay-weighting factors. Kanai et al. (2011) also focus on the delay management problem and measure the passengers' dissatisfaction based on the crowdedness in the train, the number of transfers, and the transfer waiting times. The optimization consists of creating new or breaking already existing transfers and is evaluated by a twofold simulation model to simulate the train traffic as well as the passengers' behavior. To include passengers' dissatisfaction, Tomii et al. (2005) collect a set of claim files. Depending on the type of conflicts that created the need for rescheduling, they minimize a weighted sum of relevant claim files. As a consequence, different conflicts can raise different objective functions. In their approach, the cost of the set of active claims is reduced one by one by applying some real-time interventions in the timetable.

Considering passenger service, or passengers' satisfaction is also done by Cacchiani et al. (2012b) and Shafia et al. (2012). But unlike Liebchen et al. (2010) and Vansteenwegen et al. (2006, 2007), they consider robustness as opposed to the passengers' satisfaction because of the longer planned travel times the first implies. In the approach of Shafia et al. (2012), one computes the required buffer times to achieve the desired level of robustness. Therefore, they use the stochastic behavior of disturbances with respect to the possibility of delay propagation. Cacchiani et al. (2012b), who survey different solution techniques for the TTP, distinguish two different types of objectives.

> *The nominal problem (regular TTP seen from the supplier side) is aiming at efficiency (e.g. minimizing the costs for the schedule or minimizing the total passenger travel time) while the robust problem is aiming at avoiding delay propagation as much as possible. [...] Of course, this (robustness) goes in the opposite direction of that of efficiency.* *(Cacchiani et al. 2012b)*

This quotation raises the questions of what the purpose is of short planned travel times if these travel times are unreliable? If a system is characterized by delays, is robustness not a crucial property to call that system efficient? What is a supplier with a fast but unreliable schedule that probably incurs more (operational) costs than a slightly slower but more realistic schedule? In our opinion, "robustness" and efficiency are not opposing like Cacchiani et al. (2012b) claim, but they should go hand in hand. Moreover, robustness is about acting as promised, thus being able to operate according to the published timetable. Therefore, we want to balance the inserted slack times and the speed of the timetable (Sels et al. 2013a,b; Vansteenwegen et al. 2006, 2007). Doing so, the price of robustness vanishes since, due to the balancing action, one only opts for an increase in planned travel times if it decreases the real travel times. As a consequence, the suppliers as well as the demanders benefit from it, the system becomes more reliable, and the passenger service increases.

3.2 A new definition of robustness of a railway system

In the previous section, the most important properties of robustness are identified. In spite of this, an all-integrating, acceptable and practical definition is missing. In this section, a new and all-embracing definition is formulated. The quotation from the beginning of this chapter summarizes what is to be achieved.

> *A robustness definition has to be practical in the sense that railway companies are willing to use it. This means that striving for robustness should not make the timetable too slow, and a robustness definition has to be practical to work with such that the robustness can be assessed and compared. Thus, besides a definition of robustness, an appropriate way to measure and compare timetable robustness is needed.* *(Salido et al. 2012)*

Not only the railway companies need to agree with the robustness definition, also the passengers need to be satisfied by it. By focusing on the demand side, the focus lies at the service that is offered.

By now, it is clear that an assessment of the costs and benefits of timetable slack is needed. On the one hand, time reserves are needed to avoid delays and delay propagation in which case they are *used* or were *useful*. On the other hand, slack time that is *not used* for delay compensation is *not useful* and causes extra waiting time for the passengers. An example illustrates this. Given an RTS of 10 minutes on a trip where the mean arrival delay is 3 minutes. On average, during the first 3 minutes of this supplement, the train is still driving such that these 3 minutes do not cause unnecessary waiting and are useful. During the 7 remaining minutes, the train dwells at the station which corresponds to a non-used supplement. A similar reasoning applies to transfer buffers. The part of a transfer buffer that compensates delays of the feeder train is called useful, the other part that incurs waiting for or in the connecting train, is not useful. Headway buffers do not increase the travel times. On the contrary, they are the remaining capacity (idle times) in the timetable that prevent delays from knocking on to other trains. Thus, the notion of useful or not useful does not apply to headway buffers.

Definition

Based on all collected information from the literature, observations, and discussions with the Belgian railway infrastructure manager Infrabel, our definition of robustness of a railway system can be formulated.

> ***A railway system that is robust against the daily occurring, small disturbances minimizes the real weighted travel time (RWTT) of the passengers.***

This definition states that a system is robust if and only if the average duration of all passengers' trips in practice is as small as possible. Thus, robustness is not about the travel time according to the timetable, but about the actual time it takes to arrive at your destination during operations. This includes delays and missed transfers but also the nominal travel time (NTT) and supplements.

Minimizing the real weighted travel time (RWTT) means transporting the passengers as fast as possible, even in case of small delays. Thus, according to our definition, being robust matches the idea of being able to keep the level of service under disturbed circumstances (Snelders et al. 2004).

Limited delay propagation, short absorption times of disturbances or delays, and reliable but short transfers are necessary but not sufficient for robustness. This means that, in order to be robust, a timetable should satisfy these properties. It is, however, not because these conditions are met that the schedule is robust. For example, the timetable obtained by robust optimization is not considered robust since its RWTT is not minimum. As a consequence, robustness also corresponds to attractiveness.

Observe that a used RTS is more acceptable than a non-used one because of the extra waiting time the latter causes. Furthermore, non-used supplements are more acceptable than delays since delays are not known in advance. Therefore, weights are introduced to value the differences. Section 3.3.2 elaborates further on the usage of these weighting factors.

3.3 Measuring robustness using simulation

With this definition of robustness, the most natural way to evaluate the robustness of a system is to use simulation. Therefore, a two step simulation model is built in this section. This model can be used to predict the robustness of a newly developed, microscopic planning for station areas and to evaluate the robustness of an entire network based on actual, macroscopic delay data.

Before presenting the details of the simulation tool, we elaborate upon the origin of the passenger flow information and the weights that represent the valuation of travel time. After that, an exact formula to measure the RWTT is introduced.

3.3.1 Passenger numbers

In order to measure the robustness of a timetable, passenger flow information is required. Starting from an origin-destination matrix based on the sales of season tickets[4], Sels et al. (2011a) derive the most logical routes, using a variant of Dijkstra's algorithm. For each transfer, a penalty is inserted in order to model the commuter's behavior. As a result, for each ride, dwell, boarding, alighting, and transfer action, an estimate of the amount of passengers is available. These numbers represent an importance weight for each action and allow to focus on the rush hours which are the busiest with respect to both trains and passengers. A full description of this procedure is given in Sels et al. (2011a). Notice that, in order to incorporate freight trains in the simulation, it satisfies to assign appropriate weights to each action of these trains.

During the robustness' improvement process, timetable changes are performed. Although this affects arrival and departure times, it is assumed in this dissertation that the passengers do not change trains because of these changes. If, in future work, these changes would be considered, the passenger flows can be updated by the iterative procedure of passenger routing and robustness' improvement described by Sels et al. (2011a) or we refer to Schmidt et al. (2014) for an integrated approach. Another issue that is not addressed in this dissertation is the potential increase in passenger flow due to a more robust system. When a railway system performs better, for example when the published punctuality numbers keep on improving, that system is likely to attract more passengers. Making estimations about the expected increase, however, is another field of research and thus out of scope. If in the future (or in other railway systems) more accurate information about the passenger flows becomes available, it is straightforward to use this more accurate data in our approach, instead of the current estimations.

3.3.2 Weighting factors for different types of travel time

In order to measure the RWTT, the duration of each action is weighted and counted. For example, if a passenger arrives at his destination with a delay of 2 minutes and if the weight of this type of delay is 3, this arrival delay adds $3 \cdot 2 = 6$ minutes to the RWTT. In this section, we elaborate on the weights that are used to represent the value of travel time. Determining the exact value of these weights would be a complete research project in itself and is a different field of research. Therefore, the values that will be used in this

[4] According to Vansteenwegen (2008), 85% of all passengers travels during peak hours with season tickets. As a consequence, this data gives a good idea of the mutual proportions of each train's seat occupancy.

research are based on the valuation of time in public transportation studied by Mackie et al. (2001), Savelberg et al. (2010), and Wardman (2004). Using values of time, for example, to indicate the annoyance of delays, is done in, among others, Corman et al. (2014), Goverde (1998), Sels et al. (2013b), Van Oort (2011), and Vansteenwegen et al. (2006, 2007). In the following section, the formulas to compute the RWTT as weighted sum of the durations of all real travel time components are introduced.

With the weighting factors, used and non-used timetable slack can be distinguished from delays and, in case of non-used time reserves, from the minimum necessary travel times. This has several advantages. First, minimizing the RWTT implies a minimum amount of passengers' delays. Second, it ensures that slack time will only be inserted when it is expected to be useful and thus when it decreases the RWTT. Notice that adding an RTS not only lessens delays but can also reduce the delay propagation. Thus more than 1 minute of delay can be gained by an RTS of 1 minute.

Similar to Sels et al. (2013b), four different passenger *actions* are distinguished: the start of an itinerary (boarding), dwelling, transferring, and alighting at the end of a journey. Since a ride action may influence its following action, this is not considered separately but together with, for example, the transfer after this ride action. Each action can consist of several *events*, each with their own duration or impact on the real travel time. In Table 3.1, an overview of the events per action are given. For each event, the deterministic (D) or stochastic (S) nature indicates if the occurrence and/or duration is known in advance or not. The assigned weights are added in the last column. In the next section, these weights are used to obtain the formula for the RWTT.

Four categories of weights can be distinguished. The value of 1 is used for the minimum necessary travel time and the useful supplements and buffers since these were needed to complete a particular action. A weight of 2 is assigned to the non-useful supplements and buffers. Although they slowed down the schedule more than necessary, passengers know about it in advance so it causes less harm than any kind of delays which get a weight of 3. To avoid bias, the start of a journey (boarding) gets a weight of 0 since departure delays are counted in succeeding events such as the evaluation of the usefulness of an RTS. Nevertheless, the real travel time starts at the moment of planned departure because this is the time the journey of the passengers starts. The itinerary ends at the moment of real arrival, however, like in Bates et al. (2001) and Savelberg et al. (2010), an early arrival (non-useful RTS) is penalized because it may cause extra waiting time outside the railway system, for example, when transferring to another mode of public transport.

Table 3.1: Overview of the different passenger actions, their events, nature, and assigned weights. The entries in the column *nature* indicate whether the occurrence and/or duration of the corresponding event is Deterministic or Stochastic.

action	**event**	**nature**	**weight**
board	board	D	0
	cancelation	S	3
dwell	minimum necessary ride and dwell time	D	1
	useful ride and dwell supplements	S	1
	non-useful ride and dwell supplements	S	2
transfer	minimum necessary ride and transfer time	D	1
	useful RTS and transfer buffer	S	1
	non-useful RTS and transfer buffer	S	2
	missed transfer	S	3
alight	minimum necessary ride time	D	1
	useful RTS	S	1
	non-useful RTS	S	2
	arrival delays	S	3

The difference in weight between a useful RTS and a non-useful RTS comes from their function as part of the real travel time. Where an RTS that was useful is spent driving (weight equal to one) and thus was necessary to complete the trip, a non-useful RTS extends the travel time without a reason (in this case). Moreover, when an RTS has been useful, delays are reduced and thus also the impact of these delays (in the form of delay propagation) is reduced. Therefore, non-useful slack times outweigh useful slack times. Notice that a useful RTS is no part of the NTT. Since the NTT assumes ideal situations, (external) disturbances are not considered, and therefore slack time is inserted in the schedule.

In case of a cancelation or a missed transfer, the passengers face a delay and thus these events get a weight of 3. The duration of the delay due to a canceled train or a missed transfer can be estimated based on the set of alternative trains (Dollevoet et al. 2014, 2012) or based on the period or the frequency of trains on the same line (Goverde 1998; Vansteenwegen et al. 2006). Independent of how the duration is estimated, the duration is multiplied by a weight of 3 in the RWTT, as is explained in the next section. From the moment the passengers board an alternative train, their RWTT is affected by non-used supplements or arrival delays of the alternative train.

3.3.3 Measuring robustness

The itinerary of each passenger can be represented as a sequence of the events of Table 3.1. By summing the durations of all these events, one obtains the total journey time of that passenger. Using the weights and repeating this for all passengers, the RWTT can be computed. Since each day of operation is different, the RWTT is averaged over a set of days (D). Define for all passengers p in the population P, E_p as the set of events of that passenger's journey. Then the RWTT equals

$$\mathrm{RWTT} = \frac{1}{|D|} \sum_{d \in D} \sum_{p \in P} \sum_{e \in E_p} \mathrm{weight}(e) \cdot \mathrm{duration}(d, e), \tag{3.1}$$

with $\mathrm{duration}(d, e)$ the duration of event e on day d. In (3.1), the sum ranges over all passengers. Thus, when using the term RWTT, all passengers are considered implicitly. In the remainder of this section, the same holds when talking about the nominal travel time (NTT) which is thus the total NTT of all passengers.

In the following, two variants of robustness measure (3.1) are presented. The first can be used to compare systems with a different NTT. In this case, the RWTT should be normalized to allow a fair comparison. Therefore, the normalized version of the RWTT is called the *normalized real weighted travel time ($RWTT_{norm}$)* and is measured as follows

$$\mathrm{RWTT}_{\mathrm{norm}} = \frac{1}{|D| \cdot \mathrm{NTT}} \cdot \sum_{d \in D} \sum_{p \in P} \sum_{e \in E_p} \mathrm{weight}(e) \cdot \mathrm{duration}(d, e). \tag{3.2}$$

The second variant considers the stochastic events of an itinerary. When optimizing the timetable, the infrastructure and the line planning are assumed to be fixed. As a consequence, it is just the usage of the supplements and buffers, the occurrence of delays, cancelations, and missed transfers that influence the RWTT. In order to indicate the improvement in the stochastic component of the RWTT, the *real weighted travel time extension ($RWTT_{ext}$)* is defined as

$$\mathrm{RWTT}_{\mathrm{ext}} = \frac{1}{|D| \cdot \mathrm{NTT}} \cdot \left(\sum_{d \in D} \sum_{p \in P} \sum_{e \in E_p} \mathrm{weight}(e) \cdot \mathrm{duration}(d, e) - \mathrm{NTT} \right). \tag{3.3}$$

Since it is difficult to rate the individual values for the RWTT, $\mathrm{RWTT}_{\mathrm{norm}}$, and $\mathrm{RWTT}_{\mathrm{ext}}$ of equation (3.1)-(3.3), respectively, these values should be considered relatively to a reference system. Using the RWTT or the $\text{RWTT}_{\text{norm}}$, the robustness of systems X and Y, with Y the reference system, is compared using

$$Rob_1(\%) = \frac{\text{RWTT}(X)}{\text{RWTT}(Y)} \text{ or } Rob_1(\%) = \frac{\text{RWTT}_{\text{norm}}(X)}{\text{RWTT}_{\text{norm}}(Y)}. \tag{3.4}$$

Comparing (3.1) and (3.2), one can see that there will only be a difference between using RWTT or $\text{RWTT}_{\text{norm}}$ in (3.4) if $\text{NTT}(X)$ differs from $\text{NTT}(Y)$. However, in that case, it is advisable to use the $\text{RWTT}_{\text{norm}}$. Therefore, only one robustness score is introduced. According to the definition of robustness, improving the robustness is equivalent to minimizing the RWTT. Thus, the smaller the percentage of Rob_1, the smaller the corresponding travel time of system X compared to system Y, and the more robust system X is.

For the RWTT_{ext}, the reasoning is reversed; improving the robustness corresponds to maximizing the robustness. Therefore, the comparison of two systems using the RWTT_{ext} is based on

$$Rob_2(\%) = 1 + \frac{\text{RWTT}_{\text{ext}}(Y) - \text{RWTT}_{\text{ext}}(X)}{\text{RWTT}_{\text{ext}}(Y)}. \tag{3.5}$$

Doing so, maximizing the robustness corresponds to maximizing Rob_2. The values Rob_1 and Rob_2 will be used to assess the quality of the obtained timetables in the remainder of this dissertation.

3.3.4 Simulation

Since the RWTT is hard to compute analytically, simulation is used to evaluate the robustness of different railway systems. In this section, a two step simulation model is introduced. First, conflicts and the propagation of delays are simulated using microscopic infrastructure data. Once all information about the exact arrival and departure times of trains, the usage of the supplements and buffers, etc., is known, the second step starts. In the second step, the output is generated. Next to the robustness scores Rob_1 and Rob_2, some other performance indicators are computed. This is done by simulating, event by event, the itinerary of each passenger and evaluating the evolution of train delays.

Step 1: microscopic simulation model

This part of the simulation model is used to simulate the train traffic through a station area. As input, detailed infrastructure data and timetable, routing, and platform information is used. All required data is provided by Infrabel or results from the optimization algorithm. Upon building this simulation model, some assumptions are made to reduce its complexity. As a consequence, the output needs to be interpreted with care. Nevertheless, the model is suitable for evaluating the performance of railway systems because all instances are evaluated using the same assumptions such that a fair comparison is guaranteed. A validation study using a commercial simulation package confirms this for a case study of Chapter 8.

In the discrete event driven simulation model, events are handled synchronously and stochastic influences are represented by input delays for the trains. Two types of delays are considered: delays upon entering the considered area and dwell delays at any of the $|S|$ stations, with S the set of stations. The input delays are denoted with a $(1+|S|)$-tuple representing, respectively, the delays upon arrival and the dwell delays at the stations. The size of the delays equals a predetermined value or, similar to Goverde et al. (2001), Jensen et al. (2013), Yuan (2006), and many others, is drawn from the exponential distribution using each train's real average delays $(\hat{D})$ as parameter. The latter is denoted with E from exponential, the former with $P^{(\mathrm{size})}$ from predetermined. In both cases, the number of delayed trains is added as index. Let T $(|T|)$ be the set (number) of trains, then $\left(E_{|T|/2}, P^{(0.5)}_{3|T|/4}\right)$ means that half of the trains are delayed upon arrival with the delays drawn from the exponential distribution and three-quarters of the trains gets fixed dwell delays of 0.5 minutes at the only station in the station area. To represent the daily occurring, small disturbances, only input delays smaller than 15 minutes are allowed. The upper bound is set to 15 minutes since this is the value that is used by Belgium's main passenger railway operator NMBS/SNCB as threshold for compensations against large recurrent delays[5]. Moreover, for delays of this size and larger, real-time interventions become more appropriate.

Trains enter the system at their inbound line or at the platform of departure in case of a reutilization. The trains travel from signal to signal at a predetermined speed that equals the real allowed maxima within the station area. It is assumed that the time lost by slowing down or speeding up can be approximated by a constant and only affects the travel time through the first (last) block section after (before) a stop. The size of this constant is based on the difference between the expected travel time when traveling at maximum allowed speed and the scheduled travel time for that type of train through the corresponding sections. Note that the type of rolling stock, and thus the specific acceleration characteristics, is approximated by the type of train.

[5]Source: http://www.belgianrail.be/en/customer-service/compensation-for-delays.aspx, consulted in September 2014.

Next to the assumptions about the speed profiles, some simplifications concerning the blocking times are made. In Pachl (2008), the intervals that are part of the total blocking time are being described. In the simulation model, however, three different blocking time subintervals are distinguished: the travel time through the section, the clearing time, and thirdly, an interval of constant size representing, among others, the signal processing. The first one is based on the ratio between the length of the block section(s) and the speed of the train together with the penalties for speeding up or slowing down (if applicable). The clearing time consists of the time needed for the tail of the train to leave the block section and is a function of the length of the train and its speed. The last subinterval captures the time needed for setting the signals, aligning the switches, and the time needed to release the section after the passage of a train. The minimum headway time between two trains on the same inbound line is set to three minutes, which is a commonly used threshold for trains on a common line.

Events are handled chronologically with time steps of 6 seconds. Conflicts are not predicted in advance but detected when a train approaches an already reserved block section. The conflicts are solved one by one on a first-come, first-serve basis meaning that, once detected, a conflict is solved immediately by postponing the next event of the approaching train until the estimated time the corresponding block section becomes available again. If multiple events become active simultaneously on a shared resource, extra priority rules, which are derived from practice, apply. This way, high speed trains get priority on local commuter trains and punctual trains may precede slightly delayed trains in the event list. If applicable, the number of trains within the bottleneck area is restricted by prioritizing trains that are about to leave the system compared to trains that want to enter the bottleneck area and are waiting on the open track. Based on the observation that a delayed train can be overtaken by another train outside the considered area, deviations for the planned arrival sequence at the border of the system are allowed. No real-time rerouting actions, platform changes, or cancelations of trains are made.

In the end, the real arrival times, the usage of the supplements and buffers, the locations of the conflicts, etc., all information with respect to the passage of the trains through the network is available. For every performance measure that will be evaluated in step 2, the average over 10 000 simulation runs is taken as result. As said above, the results from the simulation model need to be interpreted while keeping the assumptions in mind. Although some simplification is made, the interaction between trains is taken into account and rules from practice are applied when solving conflicts. Introducing more details in the simulation model and necessarily also within the entire developed algorithm, would complicate the computations a lot with a moderate gain in accuracy as a result. However, the question remains how external effects, such as the driver's behavior, can bias the results.

Step 2: computing the performance indicators

The output from step 1 or actual delay data can be used as input for this step. Starting from this information, together with the original (reference) timetable, the minimum necessary process times, and passenger flows from the model of Sels et al. (2011a), the passengers' travel times and other performance indicators can be computed. This is done by simulating, event by event, the itinerary of each passenger. Doing so, the RWTT and its variants $\mathrm{RWTT}_{\mathrm{norm}}$ and $\mathrm{RWTT}_{\mathrm{ext}}$ are obtained. When a missed transfer or a cancelation is detected, the number of harmed passengers is counted. Similarly, the total passengers' arrival delays are recorded. By considering a train as unit instead of a passenger, the total delays of all trains, the evolution in knock-on delays, and the percentages of *extra* or *newly delayed* trains can be found. A train is said to be extra delayed if it arrives at its terminal or leaves the system with more delays than it had upon entering. The newly delayed trains are those that were not delayed initially but got delayed during their passage through the network.

In Kanai et al. (2011), it is said that the variance reflects the fairness among passengers. For example, taking the standard deviation of a performance indicator such as the passengers' delays into account, one can distinguish between 10 times 5 minutes of arrival delays and one time an arrival delay of 50 minutes together with no delays the other 9 times. Therefore, the standard deviation of the performance indicators (between brackets) and the worst case performance with respect to the total amount of train delays are added in the result tables of the following chapters. Since the stochastic influences are equal for the different robustness measures, the standard deviation of the robustness scores is equal. The name Rob_{stdev} is used to indicate the standard deviation of the RWTT.

To test whether an improvement is significant, statistical tests are performed and a significance level of 0.05 is used. The large number of simulation runs from step 1 support the assumption that all performance indicators are (approximately) normally distributed. First the significance of the difference in variances is tested before the means are compared. Although these tests are based on the variances and not on the standard deviations, only the standard deviations are reported.

3.4 Robustness of the entire Belgian network

The idea of this section is to illustrate the applicability of the new and comprehensive definition of robustness. Based on actual delay data of the first and second quarter timetable of 2010 in Belgium (T_{Q1} and T_{Q2}), the robustness is analyzed. To obtain a fair comparison, days with bad (winter) weather or other outliers are removed from the data set. Nevertheless, it is not unlikely that some seasonal effects caused some bias. For simplicity, the delays due to missed transfers or cancelations are approximated by 15 minutes. The quarterly results are summarized in Table 3.2. In this table, one sees that the average delay per train (*train delays*, in seconds) went down with $7\% = 1 - 167/179$, while the average passengers' delays (*pax delays*, in seconds) decreased 12%. Similarly, the fraction of the passengers that are confronted with *cancel*ations or the percentage of transferring passengers who missed their *transfer* is lower for T_{Q2}. This results in a Rob_1, with T_{Q1} as reference system, of 89.7% what means that the RWTT was about 10% shorter in quarter two than in quarter one. Thus, one can conclude that the performance of the system was better during the second quarter than in the first quarter.

Like suggested by many authors, the robustness can be improved by adding time reserves in the timetable. In Tables 3.3 and 3.4, the impact of adding some running time supplement (RTS) is assessed. This is done by reducing the minimum necessary ride times for all trips by 5% or 10% without modifying the planned event times. For these new timetables, $\mathrm{T}_{5\%}$ and $\mathrm{T}_{10\%}$, no optimization or reallocation of the newly created supplements has occurred. In Table 3.3, the impact of Belgium's largest bottleneck, the North-South connection (NSC) in Brussels, is considered. Some stations in the neighborhood of the NSC are selected, and the average delays of all trains that run between the NSC and the corresponding station are computed. Not surprisingly, the punctuality of trains heading for the bottleneck ($\mathrm{D}_{\mathrm{before}}$) is better than the punctuality of those coming from the bottleneck ($\mathrm{D}_{\mathrm{after}}$). The larger the percentage of RTS, the smaller the delays and the better the robustness of the system (smaller Rob_1 value).

Table 3.2: Simulation output comparing T_{Q1} and T_{Q2}.

	train delays	**pax delays**	**cancel**	**transfer**	**Rob$_1$**
T_{Q1}	179 s	182 s	35%	14%	
T_{Q2}	167 s	159 s	27%	10%	
compare	-7%	-12%	-28%	-21%	89.7%

Table 3.3: Comparison of the train delays in some large stations before (D_{before}) and after (D_{after}) passing through the NSC in Brussels. The robustness of each system is measured for the entire Belgian network using the Rob_1 measure (3.4).

	Brussels Midi		**Leuven**		**Ottignies**		
	$\mathbf{D_{before}}$	$\mathbf{D_{after}}$	$\mathbf{D_{before}}$	$\mathbf{D_{after}}$	$\mathbf{D_{before}}$	$\mathbf{D_{after}}$	$\mathbf{Rob_1}$
T_{Q2}	191 s	268 s	146 s	301 s	130 s	238 s	100%
$T_{5\%}$	167 s	259 s	128 s	276 s	116 s	219 s	98.5%
$T_{10\%}$	142 s	251 s	114 s	253 s	104 s	202 s	97.9%

Table 3.4: The impact of time reserves on the punctuality and the robustness of some lines. The smaller the robustness scores, the more robust the corresponding timetable is.

	Line A		**Line** B		**line** C	
	pax delays	$\mathbf{Rob_1}$	**pax delays**	$\mathbf{Rob_1}$	**pax delays**	$\mathbf{Rob_1}$
T_{Q2}	185 s	100%	172 s	100%	104 s	100%
$T_{5\%}$	170 s	94.7%	157 s	98.4%	93 s	96.9%
$T_{10\%}$	156 s	90.4%	144 s	98.7%	84 s	96.9%

According to the definition of robustness, the optimal amount of time reserves is limited. This can be seen in Table 3.4. In this table, the performance on some lines is studied by comparing the passengers' delays and the resulting robustness scores along each line. Line A is a rather short line that does not pass Brussels and has a dwell supplement of several minutes halfway its trip. Line B is a long line on a busy part of the network and runs through Brussels. For this line, only the direction with terminal at the first stop after Brussels is taken into account. As such, there is not much time to recover from delays after the train passed the NSC, such that most of the supplements on the trips towards Brussels become less useful than those on the other part. At last, line C is a long line in a calmer part of the network. Unlike line A, line C has no large dwell supplements. The results from Table 3.4 have a similar pattern as those in Table 3.3. However, for line B, the size of the RTS in $T_{10\%}$ becomes too large such that too much unnecessary waiting arises.

3.5 Discussion about the presented robustness approach

This section reflects on the presented definition of robustness and on its applicability. By discussing some advantages and disadvantages of using the definition, more insight is gained in the contribution of this dissertation.

3.5.1 Usage of simulation

On the one hand, the practical usability of the newly introduced definition is an advantage since the definition indicates how the robustness of a system can be assessed: the shorter the itineraries in practice, the more robust the schedule. On the other hand, the fact that simulation is needed to evaluate the performance is a disadvantage. However, if one wants to compute other performance indicators, a simulation tool is indispensable anyway.

In order to compute all performance indicators, a lot of data is required. It is, however, this data that is the strength of the developed approach. It allows for an accurate robustness measure in which the events are valued based on the number of passengers involved. Doing so, the impact on the passengers of any change can be assessed globally and important transfers are detected.

Another advantage of simulation is that it enables easy performance evaluations on subnetworks like an entire line or a corridor as is done in Tables 3.3 and 3.4. Also the robustness of railway systems (or other public transport systems) with completely different settings can be assessed and compared.

Since optimization processes typically require many performance evaluations, using this simulation tool becomes too cumbersome within the optimization algorithm such that a quick and easy evaluation is preferred. Therefore, a substitute objective function to guide the algorithm is introduced in the next chapter.

3.5.2 Attractiveness and usability

A timetable that is robust according to the definition is attractive for both the supply and the demand side. Railway companies are pleased with a, what we call robust, timetable since it minimizes the real travel times which reduces the real costs, including delay costs. The objectives of the demand side are also met because all events that can cause hindrance or a rise in real travel time are avoided as much as possible. A fast and reliable schedule is good for the perception of the passengers. By using weights to represent the value of travel time, optimizing the robustness corresponds to improving the perception of traveling. This way, the applicability of this definition is not limited to the field of railway planning but minimizing the RWTT could be applied to other public transport networks as well.

Concerning the weights that represent the value of travel time, different valuations can be used. Nevertheless, our conclusions are based on the presented weights and we are convinced that other weights will result in similar conclusions. Further sensitivity analysis would be needed to confirm our opinion, but this is out of scope of this dissertation. However, in order to gain some more insight, the impact of setting all weights equal to 1 is assessed in Section 8.2.2.

If only the robustness of the timetable is optimized, less improvement will be found than when the entire planning process is optimized with respect to robustness. The fact that the presented definition is usable during all planning phases is another advantage[6].

At the strategic level, the usage of the introduced robustness function will result in a tendency for stations with many train lines in densely populated areas since this is necessary to obtain short real travel times. The robustness definition does not make an explicit statement about the frequency of trains. To minimize the real travel times, however, a higher frequency is needed for popular trips. Since frequency decisions are part of the line planning problem and not of the TTP or TRP, these decisions are taken by using the robustness function at the strategic level as well.

The applicability of the presented robustness definition is less straightforward for the two tactical level planning steps that are not considered in this dissertation. For the rolling stock allocation, however, one can say that the number of doors in a train is reversely related to the necessary time for alighting and boarding. Thus, selecting trains that allow fast and smooth entering and exiting helps to avoid dwell delays (J. Peeters et al. 2008).

By applying delay management strategies, one tries to minimize the real travel time in real-time. Thus, even at the operational level, the presented robustness approach is applicable.

[6] Remark, however, that the scope of this dissertation is limited to the tactical level timetabling and routing problem. Thus other problems such as the line planning (strategic level) or the development of real-time control systems (operational level) are not considered here.

3.5.3 Scientific value

From a scientific point of view, this definition is interesting since it captures all notions of robustness of other authors; minimum real travel times imply minimum delays, minimum propagation of delays, reliable transfers, etc. Vice versa, all the individual goals relate to bringing the passengers as fast as possible and in a reliable way to their destination, which is exactly what we claim to be robustness.

The first results of applying this definition were successful. First of all, Vansteenwegen et al. (2006, 2007) studied the inclusion of an RTS on the feeder train's trip towards the transfer station by assessing its impact on the real travel time for all passengers. Second, there is the technique of retime and reflow (Sels et al. 2011a) to construct a timetable while accounting for the passengers' route choice behavior. In Sels et al. (2013a,b, 2011b), a mathematical model is built in which the RWTT is included in the objective function. Therefore, the passengers' real travel time is analytically derived as a stochastic function. No routing decisions and microscopic infrastructure details are considered. Nevertheless, a timetable for the whole Belgian network could be computed in a couple of hours and, among others, the chance of missing a transfer is reduced significantly.

3.6 Conclusions

A comprehensive definition of robustness is the central topic of this chapter. Based on the properties of robustness that are used in literature, the essence of what railway robustness is about is detected. This enabled us to define a robust system as a system in which the passengers' real weighted travel time (RWTT) is minimum. The applicability of this definition is proven by measuring the robustness of a timetable for the whole Belgian network using actual data. Therefore, a simulation model is developed and the necessary formulas to compute the RWTT are introduced. From the simulation results, the negative impact of the NSC on the punctuality became clear.

With the presented robustness definition and the whole discussion around it, an answer is provided to the first research question about how to define the robustness of a railway system.

Chapter 4

Optimizing the robustness of a railway system

A major challenge for further research is to find a better integration between the models developed for solving the timetabling step [...] and the models developed for routing trains through stations [...]. A better integration between these models [...] may be beneficial for the quality of the finally obtained timetable, in particular for its robustness.

Kroon et al. (2008a)

The results of the simulation study of Section 3.4 indicate that traversing Brussels goes hand in hand with lots of extra delays. This conclusion can be generalized to other bottlenecks and, in particular, to large and complex station areas (Yuan 2006). That is why we developed an algorithm to improve the robustness in such station areas. In this chapter, the framework of this algorithm, as well as the objective function that is used to guide it, is explained. Next to that, the case study of the North-South connection (NSC) in Brussels is introduced and an overview of the main assumptions is given.

This chapter is based on Dewilde et al. (2013, 2014).

4.1 Framework of the algorithm

This dissertation focuses on tactical level planning problems, and more specifically on the train routing problem (TRP) and the train timetabling problem (TTP). Acknowledging the statement of Kroon et al. (2008a) from the beginning of this chapter, we developed an algorithm that integrates the optimization of the route choices, the ordering and/or timing decisions, and performs platform changes, all in an iterative way. At the start, a reference timetable serves as initial solution.

The algorithm itself consists of three modules. An overview of the framework is given in Algorithm 4.1 and Figure 4.1 is used to visualize the interaction between the modules. The first improvement comes from calling the *routing module* to solve the TRP. After that, the *timetabling module* follows and considers the TTP. If the timetabling module did not find any improvement, the *platforming module* is started in order to test the impact of platform changes (downward arrow in Figure 4.1). Otherwise, the leftward arrow leads to the routing module that is solved again right after the timetabling module has finished.

The TRP is solved to optimality and the timetable and platform allocations are improved using metaheuristic techniques[7]. Moreover, both the timetabling module and the platforming module use the timetabling module as a subroutine in which case it is called the *internal timetabling module* (shaded arrows in Figure 4.1). The iterative procedure of the modules stops if no improvement is found for a number of consecutive iterations ($iter^{\max}$). In the following chapters (5, 6 and 7), each of the modules is discussed one by one together with the details for the stop criteria.

Algorithm 4.1 Framework of the developed algorithm

input: infrastructure data and reference timetable
while number of consecutive non-improving iterations $\leq iter^{\max}$ **do**
  solve the train routing problem (TRP)
  apply tabu search (timetabling module)
  if timetabling did not yield improvements
    then start the platforming module

[7]Several studies have shown that a full integration of routing, timetabling, and platforming within an exact algorithm is not appropriate for real-life case studies (Caimi et al. 2005; Kroon et al. 2007b). Therefore, we decided to work with separate modules and opted for metaheuristics when the currently available technology proved to be inadequate to solve practical instances like ours. We come back to this issue in Chapter 6.

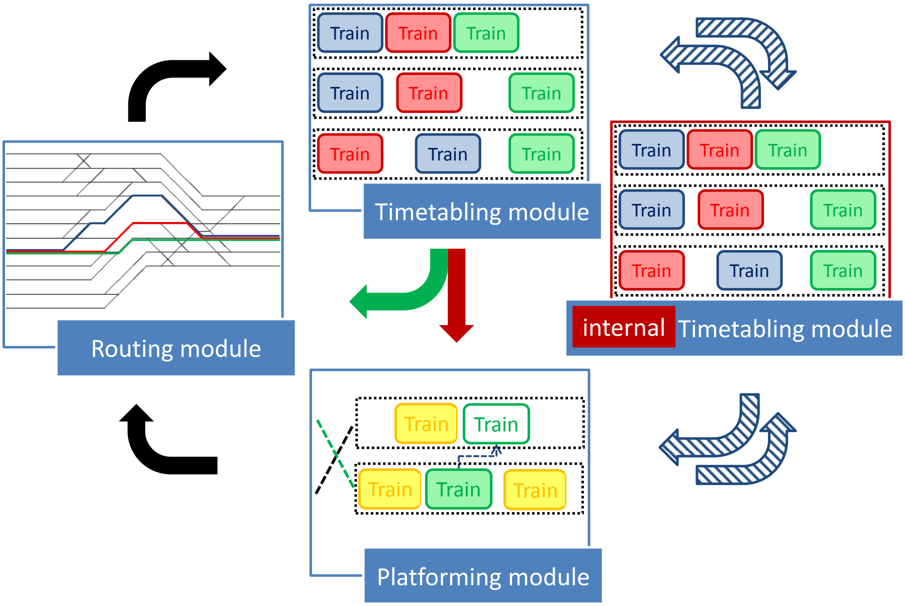


Figure 4.1: Visualization of the interaction between the modules of the algorithm. In the routing module, each train gets assigned one route through the network. After that, the time between two trains (represented by the spaces between the trains) is modified in the timetabling module. If no improvement is found, a train is assigned to another platform (downward arrow). Since some timetable changes can be required to make this reassignment, the internal timetabling module is used within the platforming module (as well as within the timetabling module for some other changes). After the platforming module or after improvement is found in the timetabling module (leftward arrow), the routing module is called.

4.2 Objective function of the algorithm

All three modules of the developed optimization algorithm strive for the same goal: improve the robustness. Since evaluating the change in robustness for each and every potential adjustment of the system would require far too much computation time, a substitute objective function is used throughout the entire algorithm. This function considers the spreading of the trains which is measured by the minimum time span between any two trains. Thanks to this function, quick quality evaluations can be made.

Similar to the assumption of Caimi et al. (2009a) that trains travel through their condensation zones at maximal speed, thus without time reserves, we argue that capacity restrictions do not allow for extra supplements in the large and busy station areas that are considered in this research. Thus, according to the definition of robustness in Kroon et al. (2008a), see Section 3.1.2, headway buffers are the only option to enhance the robustness in such areas. Moreover, as discussed in Section 3.2, no distinction is made between useful and non-useful headway buffers. Thus, redividing these buffers does not directly influence the RWTT. Indirectly, however, less knock-on delays should arise. That is why the spread of the trains is optimized. Notice that this is a commonly used technique to indirectly improve the robustness (Caimi et al. 2005; Kroon et al. 2008d; Salido et al. 2012). Since spreading the trains is not our main goal, but robustness is, the impact on the robustness of the improved spreading is measured afterwards using the simulation module.

Block sections and minimum time spans

Before presenting the details of the objective function, some background about block sections is needed. In general, a railway network is divided in block sections that are bordered by signals. Block sections are the basis of the railway's safety system. A simple rule of thumb is that two trains may not use the same tracks or switches within a block section at the same time. In Figure 4.2, a network with two parallel tracks that are connected by switches is depicted. The (two-way) signals in the network are indicated by ○⊣⊢○. Each subnetwork that is bordered by signals is considered a block section. In this figure, three block sections (bs_1, bs_2, and bs_3) are indicated. The tracks belonging to block section bs_1 are $(3, 5), (4, 6), (5, 6), (5, 7)$, and $(6, 8)$. Block section bs_2 consists of the two parallel tracks $(7, 9)$ and $(8, 10)$.

Trains run from block section to block section and the interval during which a block section is reserved for a certain train is called the blocking time (see 3.3.4). When a train approaches a block section, all tracks on its route within that block section are reserved at once. If two trains claim the same resources (a track or a switch) within one block section, the second train may only enter this section

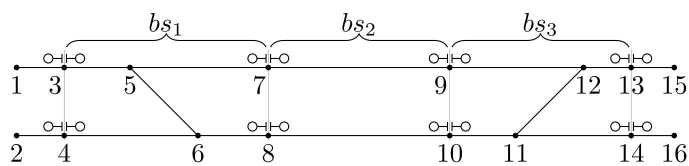


Figure 4.2: Example of a network with 3 block sections. The signs ○⊣⊢○ are used to indicate the (two-way) signals.

after it is released by the first train. For example, in Figure 4.2, a route via nodes 3, 5, 6, and 8 and a straight route from node 3 to node 7 cannot be used simultaneously since track $(3, 5)$ is used in both routes. The blocking times of a train that stays on the upper track between node 1 and node 15 and a train that runs from node 2 to node 16 on the lower track can be overlapping since no tracks or switches are common in both routes.

Let T be the set of trains and R the set of routes that trains can follow through the station area. In the following, the term *trainroute* is used to indicate the combination of a train ($t \in T$) and its route through the station area ($r_t \in R_t$), with $R_t \subset R$ the set of candidate routes for train t. With this notation, (t, r_t) is a trainroute. Denote the set of block sections with BS. The notation $B^{bs}_{(t,r_t),(t',r_{t'})}$ is used for the headway buffer between trainroutes (t, r_t) and $(t', r_{t'})$ in block section $bs \in BS$. This is the time between the reservation of bs by train t and that of train t'. It is assumed that $B^{bs}_{(t,r_t),(t',r_{t'})} = \infty$ if no resource is shared by the two trains within bs. The minimum time span between trains t and t' with their routes r_t and $r_{t'}$, respectively, is then computed as

$$B_{(t,r_t),(t',r_{t'})} = \min_{bs \in BS} B^{bs}_{(t,r_t),(t',r_{t'})}. \tag{4.1}$$

The block section bs^* for which $B^{bs^*}_{(t,r_t),(t',r_{t'})} = B_{(t,r_t),(t',r_{t'})}$ is called the *critical block section*. Note that the critical block section depends on the timing and the routing of both trains and that the critical block section might change when the routing or the timing changes. We assign a cost to each minimum time span as follows

$$C_{(t,r_t),(t',r_{t'})} = \begin{cases} 15 & \text{if } B_{(t,r_t),(t',r_{t'})} = 0 \text{ (conflicting)}, \\ 1/B_{(t,r_t),(t',r_{t'})} & \text{if } B_{(t,r_t),(t',r_{t'})} < B^{\max} \text{ minutes}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

The parameter $B^{\max}$ denotes the shortest duration of a minimum time span that is considered insensitive to conflicts. Like Cacchiani et al. (2012a), who claim that the utility of longer slack times is questionable, a threshold of 15 minutes is used for $B^{\max}$. The cost of a conflict is set equal to 15 because a precision of 0.1 minutes is used for computing the headway buffers causing $C_{(t,r_t),(t',r_{t'})}$ to be smaller than 15 if $B_{(t,r_t),(t',r_{t'})} > 0$. C is used as cost in the *spreading* objective function

$$\text{Minimize} \sum C_{(t,r_t),(t',r_{t'})}. \tag{4.3}$$

Using this function has several advantages. First of all, due to the usage of the reciprocals in (4.2), smaller time spans correspond to higher costs and there is a decreasing marginal effect of increasing a time span. Second, since the sum ranges over all pairs of trainroutes, one can assess the impact of each timetable change that causes increased and decreased time spans, and third, it provides a fast and easy way to evaluate a timetable during the search for improvements. Therefore, this function is used in the algorithm to guide the improvement process.

4.3 Robustness and the spreading objective function

A difference between the spreading objective function and the robustness measure relates to the usage of passenger numbers. Robustness is about the passengers' RWTT, whereas the objective function optimizes the minimum time span between each trainroute pair without considering passengers. This discrepancy has two reasons. First, if a train gets delayed, all current and future passengers on that train are infected. Thus, when including weights based on passenger flows in the objective function, one should account for all potential passengers. Second, since the number of passengers on a train changes discontinuously from trip to trip, passenger weighted objective functions become discontinuous if the critical block sections change due to timetable modifications along the improvement process. As a consequence, an increased minimum time span can incur a larger spreading cost if the passenger weights are higher in the new critical block section.

Due to this discrepancy in the usage of passenger numbers, improvements in objective function value do not always correspond to gains in robustness. Figure 4.3 illustrates this. In order to generate this figure, simulation was used to determine the Rob_1 and Rob_2 robustness scores as well as the amount of knock-on delays of the currently best solution per iteration (according to the spreading objective function). The initial reference values are set to 100%. Where the spreading cost and the amount of knock-on delays are scaled based on the left Y-axis, the two robustness scores are scaled using the right axis. Although the objective function value is only decreasing in the course of the algorithm, the other performance indicators fluctuate. We come back to this Figure in Chapter 7.

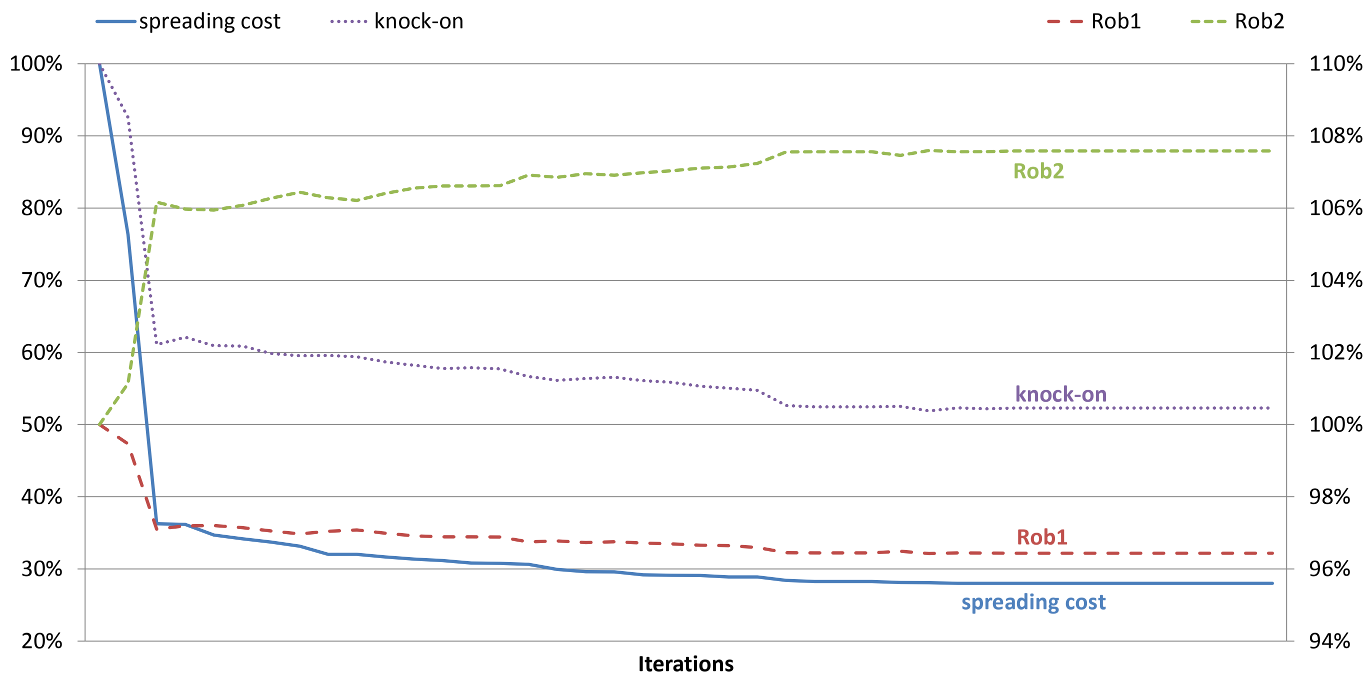


Figure 4.3: Evolution of some performance indicators in the course of the algorithm. Per iteration of the algorithm, the currently best solution is evaluated based on the objective function (*spreading cost*), the Rob_1 and Rob_2 robustness scores, and the amount of *knock-on* delays. The dashed curves for the robustness scores are scaled based on the right Y-axis, the others on the left Y-axis.

4.4 Assumptions

Several assumptions are made when developing the algorithm. In Section 3.3.4, an overview of the assumptions that are made for the simulation model is given. These premises were related to the minimum necessary travel time, the blocking times, and the minimum headway times between two trains on open track sections. For the algorithm, the same assumptions are made.

Similar to the arguments in Zwaneveld (1997) about approximating the length of the trains by a certain maximum length due to the lack of rolling stock information during the TRP (and TTP), we assume that trains have a fixed length, run at the maximally allowed speed, and use compensations for slowing down or speeding up before or after a dwell action.

The available input data was just precise enough to estimate the length of routes by the inter-signal distances. Therefore, routes with the same set of signals have equal lengths and thus require equal running times. These running times are computed as follows. Denote with σ_1 and σ_2 the two signals of a block section and let $dist(\sigma_1, \sigma_2)$ be the distance between them. If $v_t^{\max}$ is the maximally allowed speed of train t in that block section, then the running time through this block section for (the head of) this train equals

$$\left\lceil \frac{dist(\sigma_1, \sigma_2)}{v_t^{\max}} \right\rceil_{0.1\,\mathrm{min}}. \tag{4.4}$$

The time until the tail of the train leaves the block section is computed likewise. The rounding action is due to the precision of the computations up to 6 seconds.

At the start of the algorithm, the routing module searches for the routing solution with the smallest spreading cost. Doing so, no changes in arrival and departure times of trains are allowed. Therefore, it is assumed that at least one conflict-free routing solution for the reference timetable exists.

The last assumption is about the impact of timetable changes on the outside of the considered network. Although feasibility is maintained within the station area itself, as discussed in Section 1.4, we assume that possible conflicts that arise at other locations in the network are not impossible to solve, and therefore, these are left aside.

4.5 The North-South connection (NSC) case study

The algorithm that is developed in this dissertation focuses on large and complex station areas. The network of Brussels with its NSC is a typical example of such

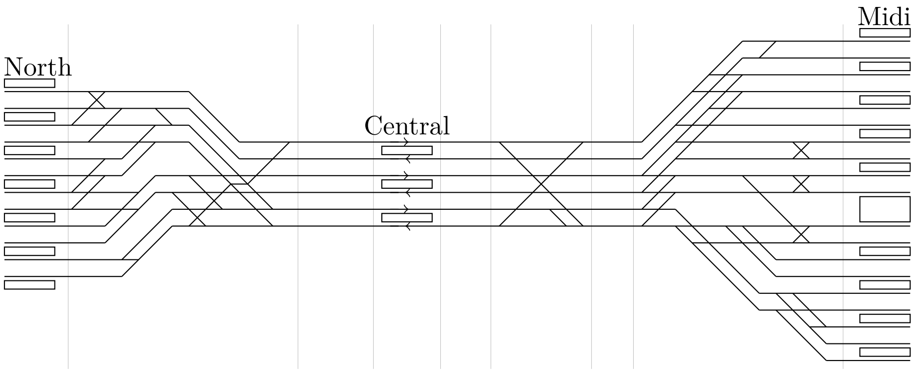


Figure 4.4: Infrastructure details of the North-South connection (NSC) between the stations North and Midi of Brussels. The sectors are bounded by the gray lines which represent the locations of the signals.

a station area. This compact and highly used network is the beating heart of the Belgian railway system. It truly is an interesting case study since it contains three of the country's four busiest stations regarding passenger numbers (Sels et al. 2011a; Stynen 2010). It includes the largest station with respect to the number of platforms and a true, physical bottleneck since the 19 (through) platforms of station Midi are connected with the 12 platforms of the North station through a 6-track tunnel, the North-South connection (NSC) with the Central station. Figure 4.4 gives a detailed overview of the infrastructure connecting the three stations. The planned travel times North-Central and Central-Midi are 3 minutes. Next to the tracks towards the Central station and a shunt yard, each of the outer stations has four in- and outbound orientations. Trains run from all over the country towards Brussels, forming a crisscross of lines with many merging and intersecting routes in the station area as a consequence. The fact that the orientation of the tracks in the NSC is alternating intensifies this.

In total, there are up to 90 trains per (peak) hour that visit the Brussels' station area. This makes that the capacity utilization is nearly saturated[8]. Some trains turn around at station Midi and do not run through the NSC, but nearly all of them pass or dwell at the Central station. This forces the planned dwell time to be limited to one minute which is nearly always insufficient in practice[9]. Next to

[8]During the construction of the nearly two kilometer long tunnel, up to 700 trains per day were expected in the NSC. Nowadays, this number is largely exceeded (J. Peeters et al. 2008).

[9]The Central station was initially designed for maximally 50 000 daily passenger movements. However, the last passenger counts in 2009 resulted in more than 72 000 boarding actions during a workday (Stynen 2010). Together with some rolling stock types that are not well

the capacity issues, also the grids at both sides of the tunnel are conflict-prone. Because of the interweaved routes, the number of trains that share resources is very high. As a consequence, the set of hindered trains often is larger due to a conflict in one of the grids than due to an extended stop. Nevertheless, the delays that arise from a small disturbance anywhere in the centrally located Brussels' area easily spread out in space and time. Therefore, it is expected that an optimized local schedule and an appropriate routing through the station area will help to improve the performance on the whole railway network.

The timetable that is used as input for the algorithm and as reference situation in the simulation module is used for many years in practice and is improved step by step in the course of the years. Only the 80 trains that are scheduled to run through the NSC between 7 and 8 AM are considered for this case study.

In Schaafsma et al. (2007), it is argued that no transfers are scheduled in the bottleneck area of Schiphol in the Netherlands. Doing so, synchronization actions that slow down the traffic through the bottleneck are avoided. In Brussels, there is a high frequency of trains in each direction such that the extra waiting time caused by a missed transfer is supposed to be limited. As a consequence, no transfers are scheduled and no data about it is recorded for the Brussels' area. Remark, however, that due to the large amount of lines that meet in Brussels, many passengers make a transfer there. Similar to what is done in practice, no transfers are considered during the optimization and simulation for the NSC case study. Nevertheless, a list of guaranteed transfers could easily be taken into account in the algorithm. If a timetable change would make a transfer infeasible, the corresponding feeder or connecting train should also be shifted.

The network of the NSC case study is restricted to Figure 4.4. In comparison with Figure 1.1, only the inner part of the entire Brussels' area is considered. For the computational results in Chapters 5-7, the NSC case study is used. In Chapter 8, however, the computations are repeated for the entire Brussels' station area. Next to the inclusion of the grids at the outsides of the stations Midi and North and the extension of the network until the beginning of the open track sections or the entrance of the shunt yards, the infrastructure is modeled more in detail. The difference is twofold. First, in the network of Figure 4.4, the block sections are approximated by sectors that are bordered by collinear signals. Second, in contradiction with (4.4) that is used for the case studies of Chapter 8, the travel times through the sectors are predefined and independent of the followed routes.

---

suited for many boarding or alighting passengers, the necessary dwell times can sometimes rise to three minutes during peak hours (J. Peeters et al. 2008).

4.6 Conclusions

In this chapter, the idea behind the answer to the second research question about how to deal with the limited capacity in the North-South connection (NSC), is revealed. By integrating the scheduling and routing of trains through the network, a better usage of the available capacity is quested for. The spreading objective function aims at minimizing the spreading cost and allows an assessment of the increased and decreased time spans of each change. After considering the assumptions that are made for the algorithm, the NSC case study is introduced. The specific properties that illustrate the relevance of the second research question are discussed and the differences with the more detailed and more extended case study of Chapter 8 are explained.

Chapter 5

The train routing problem

Railway stations often turn out to be the main source of delays in a dense railway system due to their limited routing possibilities. Therefore, investigating robust routings of trains through the stations is highly relevant for improving the punctuality of a railway system.

Kroon et al. (2008c)

The train routing problem (TRP) is about finding a path for each train through the considered network. More precisely, given a set T of trains and the set R of all possible routes through a station, a solution of the TRP is an allocation of exactly one route $r \in R$ to each train $t \in T$ in a conflict-free way.

Upon solving the TRP, the arrival and departure time of each train as well as the assigned platforms in each station are considered unchangeable. Therefore, it is assumed that at least one conflict-free routing solution exists for the active timetable.

This chapter starts with presenting the details of the routing module. After presenting the computational results of applying the routing module to the North-South connection (NSC) case study of the Brussels' area, the differences with the most related solution techniques from the literature are discussed.

This chapter is an extension of Dewilde et al. (2014).

5.1 The routing module

To solve the TRP, the routing module consists of three steps. The first step is about the set of routes which gets linked with the set of trains in step 2. After a preprocessing phase, the TRP can be solved to optimality. This is the content of the third step.

5.1.1 Step 1: route dominance

In the first step of the routing module, the set R of all possible routes is determined. A route is a sequence of (platform-)tracks and switches from one end of the network (station area) to the other end. For each combination of an origin and a destination, alternative routes may exist. Consider, for example, the single block section network of Figure 5.1. At the top of this figure, the network is drawn. Below, each of the five routes through this network are depicted and the nodes along each route are named. The first two routes (r_1 and r_2) connect node 1 with node 11. From the discussion about block sections in Section 4.2, we know that routes r_1 and r_2 cannot be used simultaneously, and that the blocking times of two trains following routes r_1 and r_3 can be overlapping since no resources are common. Comparing the two routes between node 1 and node 11, route r_2 seems to be superfluous; since route r_2 makes a detour compared to r_1, replacing r_2 in any solution by r_1 will never cause a worsening in the spreading objective function value. As a result, route r_2 can be removed from R. However, if the network consists of multiple block sections, things change. This is the case in Figure 5.2 where the simultaneous usage of r_1 and r_2 by two trains in opposing directions is allowed under certain conditions. Thus, pruning detour routes does not always work and a more formal preprocessing rule is needed.

Define a *link* as a track section bounded by switches, signals, or a combination of these. According to this definition, a link belongs to only one block section $bs \in BS$. Let r be a route and $l \in L^{bs}$ be a link with L^{bs} the set of links within block section bs, then $l_r \in L^{bs}$ denotes a link on route r ($l_r \in r$) and, vice versa, r_l is used for a route that uses link l ($l \in r_l$). The symbol $\leftrightarrow_{bs}$ is used to indicate compatible routes in bs. Compatible routes are routes that can be followed by trains with overlapping blocking times at block section bs. For example, routes r_1 and r_3 in Figure 5.1 are compatible. The set BL_r^{bs} of blocked links that cannot be used in parallel with route r within block section bs is defined as

$$BL_r^{bs} = \left\{ l \in L^{bs} : \nexists r_l \in R : r \leftrightarrow_{bs} r_l \right\}. \tag{5.1}$$

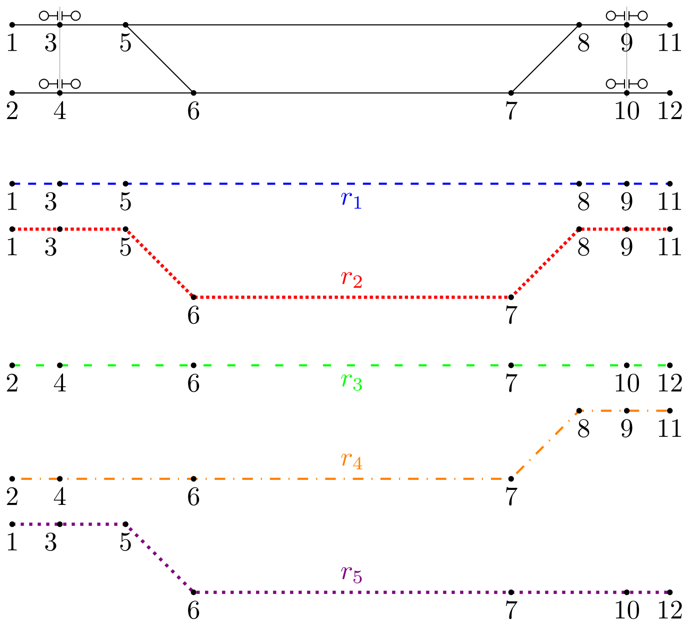


Figure 5.1: Example of a detour route within one block section. In this network with two parallel tracks and some switches to connect these tracks, there are five routes to travel from one end to the other end. Each of these is visualized individually below the network.

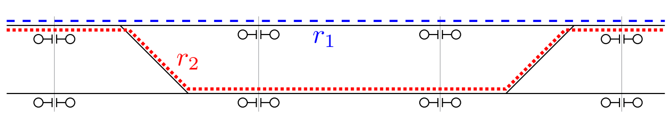


Figure 5.2: The impact of signals on detours. A train that comes from the left and uses route r_2 can, under certain conditions, be simultaneously in the network with a train coming from the right and following route r_1.

Thus $l \in BL_r^{bs}$ if there exists no route that uses l and is compatible with r. An example helps to clarify the concept of blocked link sets. Let bs be the block section in the middle of the network in Figure 5.1. In this network, the blocked link set of route r_1 in block section bs ($BL_{r_1}^{bs}$) consists of links $(3,5),(5,8),(8,9),(5,6)$ and $(7,8)$. The first three links are those along the route. The last two links, $(5,6)$ and $(7,8)$, are also blocked because no train can use them at the same time as route r_1 is reserved by another train. Links $(1,3)$ and $(9,11)$ do not part $BL_{r_1}^{bs}$ because they do not belong to L^{bs}. Since $r_1 \leftrightarrow_{bs} r_3$, the links of r_3 are not blocked either. Thus

$$BL_{r_1}^{bs} = \{(3,5),(5,8),(8,9),(5,6),(7,8)\}. \tag{5.2}$$

The blocked link set of route r_2 in bs differs from the one of r_1. As no route is compatible with route r_2,

$$BL_{r_2}^{bs} = L^{bs}. \tag{5.3}$$

Based on the set of blocked links for each block section, the entire *blocked link set* of route r can be constructed as follows

$$BL_r = \bigcup_{bs \in BS} BL_r^{bs}. \tag{5.4}$$

Before showing the usefulness of the blocked link sets, the way to construct BL_r for each route r is discussed. The necessary steps to obtain BL_r^{bs} for each block section bs are presented in Algorithm 5.1 and equation (5.4) is used to

Algorithm 5.1 Procedure to construct the blocked link sets

input: a route r and a block section bs
$BL_r^{bs} = \emptyset$
(i) **for all** $l \in L^{bs}$ **do**
  if $l \in r$ **then** $BL_r^{bs} \leftarrow BL_r^{bs} \cup \{l\}$
(ii) **for all** $i \in N^{bs}$ **do**
  if $i \in r$ **then** $BL_r^{bs} \leftarrow BL_r^{bs} \cup E(i)$
(iii) **for all** $l = (i,j) \in BL_r^{bs}$ **do**
  if $l \notin r$ and $i, j \in N_{in}^{bs}$ **then**
    for all $l' = (i',j') \in E(i) \cup E(j)$ **do**
      if $l' \notin BL_r^{bs}$ and $\nexists r_{l'} \in R : r \leftrightarrow r_{l'}$ **then**
        $BL_r^{bs} \leftarrow BL_r^{bs} \cup \{l'\}$
  (iv) **if** $E^+(i') \subset BL_r^{bs}$ **then** $BL_r^{bs} \leftarrow BL_r^{bs} \cup E(i')$
    (Analogue for $E^-(i'), E^+(j')$ and $E^-(j')$)

get BL_r. To continue, some notation is needed. Each link l can be denoted as (i, j) with i and $j \in N^{bs}$, the set of nodes within bs. A node represents a switch or a signal. The set of nodes different from signals equals N^{bs}_{in}. To indicate that route r passes by node i, $i \in r$ is used. The links surrounding node i form the set $E(i)$ and $E^-(i)$ $(E^+(i))$ is the set of incoming (outgoing) links of node i. Independent of how the trains are moving (from left to right or vice versa), incoming links of node i are defined as those on the left hand side of i, outgoing nodes are those on the right hand side of i. As a result, for each $i \in N^{bs}_{in}$ and $r \in R$ holds that if $i \in r$ and the predecessor of i in r belongs to $E^-(i)$ $(E^+(i))$, then its successor must belong to $E^+(i)$ $(E^-(i))$.

Figures 5.3-5.5 are used to illustrate the BL^{bs}_r construction procedure of Algorithm 5.1. In Figure 5.3, the infrastructure within one block section, say bs, and the route r that passes through this block section are drawn. Node 5 is considered a special switch since it only allows to travel towards node 12 coming from node 2 and not from node 1. (i) The first step in Algorithm 5.1 is the addition of all links of r to BL^{bs}_r. (ii) After that, all links that surround a node of route r are added to the blocked link set within section bs. In Figure 5.4, all these nodes are filled and the links that are added in this step are indicated with the dotted links. (iii) Then each blocked link $l = (i, j)$ that is no part of route r and does not border the block section's boundary is studied one by one in the third step. For each of the surrounding links of i or j, say l', the blocking property $\nexists r_{l'} \in R : r \leftrightarrow_{bs} r_{l'}$ is tested. If no such route $r_{l'}$ exists, l' cannot be used simultaneously with r and is thus blocked. Therefore, it is added to BL^{bs}_r. (iv) If all incoming or outgoing links of a certain node become blocked, all surrounding links of that node are also blocked. Step (iii) and (iv) are repeated until all blocked links, also the newly added ones, are considered.

The final blocked link set of route r in bs is shown in Figure 5.5. Since blocked link $(6, 8)$ is no part of r and neither 6 nor 8 is a border node, its surrounding, non-blocked links $(3, 6)$ and $(5, 6)$ are considered in step (iii). The figure clearly shows that there exists no route that is compatible with route r and that passes along $(3, 6)$ or $(5, 6)$. Therefore, both can be added to the set of blocked links. Then the algorithm continues and at a certain point, blocked link $(5, 6)$ becomes active $(l = (5, 6))$. Links $(2, 5)$ and $(5, 12)$ do not satisfy the blocking property because of the route from 2 to 15 via 5 and 12, but link $(1, 5)$ does since the infrastructure limitations do not allow to follow $(5, 12)$ after $(1, 5)$. Since the only incoming link of node 11 is blocked, no compatible route can pass along this node. As a consequence, links $(11, 12)$ and $(11, 13)$ are also blocked. The entire set of blocked links of route r is indicated in Figure 5.5.

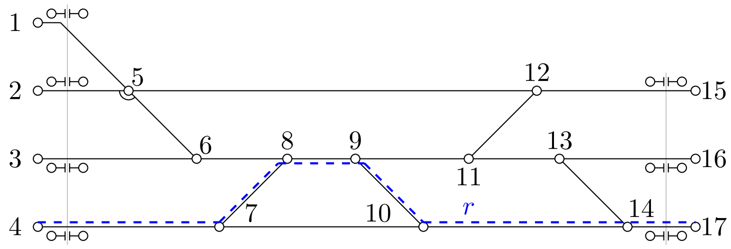


Figure 5.3: Construction of the set of blocked links of route r: initialization. The infrastructure at node 5 does not allow to travel from node 1 towards node 12. All other combinations are possible.

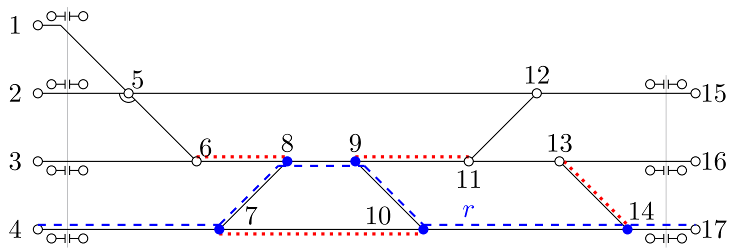


Figure 5.4: Construction of the set of blocked links of route r: intermediate result. The nodes on r are filled and the links surrounding them are indicated with the dotted lines.

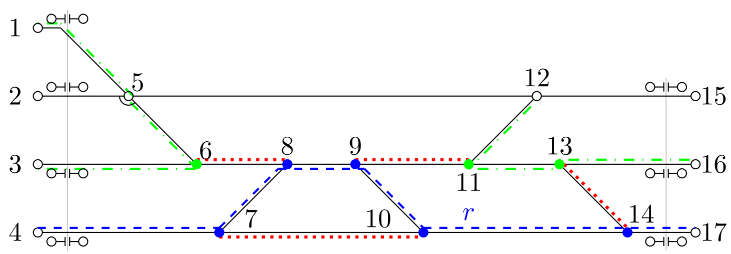


Figure 5.5: Construction of the set of blocked links of route r: final result. The entire set of blocked links is highlighted. Notice that only the route from node 2 towards node 15 via nodes 5 and 12 is compatible with r such that all other links are part of BL_r^{bs}.

Now, it is time to show the usefulness of the blocked link sets.

Theorem 1. *Let r and r' be two routes between the same origin and destination and with the same set of signals. If BL_r is a strict subset of $BL_{r'}$ ($BL_r \subsetneq BL_{r'}$), then r route dominates r'.*

A route r *(route) dominates* another route r' with the same origin and destination and the same set of signals, indicated with $r \preccurlyeq r'$, if using r instead of r' does not worsen the objective function value (4.3). Let R_t be the set of routes that are suitable for train t, then

$$r \preccurlyeq r' \Leftrightarrow \begin{array}{l} \forall t \in T : r, r' \in R_t \text{ holds that} \\ \forall(\bar{t}, \bar{r}) \in T \times R : C_{(t,r),(\bar{t},\bar{r})} \leq C_{(t,r'),(\bar{t},\bar{r})}. \end{array} \tag{5.5}$$

Note that $C_{(t,r),(\bar{t},\bar{r})} \leq C_{(t,r'),(\bar{t},\bar{r})}$ corresponds to $B_{(t,r),(\bar{t},\bar{r})} \geq B_{(t,r'),(\bar{t},\bar{r})}$ or $B_{(t,r),(\bar{t},\bar{r})} \geq B^{\max}$. From (5.5), one sees that when a route is dominated, it can be removed from the set of candidate routes without running the risk of losing feasibility or worsening the optimal spreading cost.

To illustrate this theorem, consider again the routes in the network of Figure 5.1. As example of the definition of blocked links sets, the blocked link sets of routes r_1 and r_2 are computed in (5.2) and (5.3). Since $BL^{bs}_{r_1} \subsetneq BL^{bs}_{r_2} = L^{bs}$, the theorem gives that r_1 dominates r_2. This matches the observation from the beginning of this section.

The proof of Theorem 1 is based on contraposition.

Proof. Let r and r' be two routes between the same origin and destination and with the same set of signals such that BL_r is a strict subset of $BL_{r'}$. Take an arbitrary $t \in T$ for which $r, r' \in R_t$. Assume that r does not dominate r'. Following (5.5), there exists a trainroute $(\bar{t}, \bar{r}) \in T \times R$ for which $C_{(t,r),(\bar{t},\bar{r})} > C_{(t,r'),(\bar{t},\bar{r})}$. According to (4.1) and (4.2), there is a block section $bs \in BS$ such that

$$0 \leq B^{bs}_{(t,r),(\bar{t},\bar{r})} < B^{bs}_{(t,r'),(\bar{t},\bar{r})} \tag{5.6}$$

with

$$B^{bs}_{(t,r),(\bar{t},\bar{r})} < B^{\max}. \tag{5.7}$$

From (5.6) and (5.7), it follows that for all links of $\bar{r}$ ($l_{\bar{r}}$) holds that $l_{\bar{r}} \notin BL^{bs}_{r'}$ since otherwise (5.1) states that no $\hat{r}_{l_{\bar{r}}}$ exists that is compatible with r', while (5.6) implies that $\bar{r}$ is compatible with r' and $l_{\bar{r}} \in \bar{r}$. If the blocking times of t and $\bar{t}$ overlap, the compatibility follows directly from $0 < B^{bs}_{(t,r'),(\bar{t},\bar{r})}$. In the other case, (5.7) means that $\{l \in L^{bs} : l \in r \text{ and } l \in \bar{r}\} \neq \emptyset$, and (5.6) gives that $\{l \in L^{bs} : l \in r' \text{ and } l \in \bar{r}\} = \emptyset$ since otherwise the buffers would be equal.

Thus for all $l_{\bar{r}}$ holds that $l_{\bar{r}} \notin BL^{bs}_{r'}$. Because $BL^{bs}_{r} \subsetneq BL^{bs}_{r'}$, it is sure that $l_{\bar{r}} \notin BL^{bs}_{r}$. As a consequence, $\bar{r}$ and r have no resources in common in bs. This implies that for all t and $\bar{t}$, $B^{bs}_{(t,r),(\bar{t},\bar{r})} = \infty$ what contradicts with $B^{bs}_{(t,r),(\bar{t},\bar{r})} < B^{\max}$ and concludes this proof. □

An important remark relates to the symmetry of the block sections. If the locations of the signals are not the same in both orientations, the block sections are not symmetric. As a consequence, the route dominance should be applied per direction. Consider, for example, the network of Figures 5.1 and 5.2. If in one orientation, the signals are located like in Figure 5.1 and for the other orientation like in Figure 5.2, then route r_2 is dominated for the first orientation only. Therefore, set R should be duplicated before the route dominance starts.

A stronger version of route dominance would be strict route dominance. Route r strictly dominates route r' if there exists at least one routing solution of which the spreading cost improves when r' is replaced by r. This corresponds to the requirement that there must be a trainroute $(\hat{t}, \hat{r})$ that allows at least one strictly larger minimum time span.

$$\exists(\hat{t}, \hat{r}) \in T \times R : B_{(t,r),(\hat{t},\hat{r})} > B_{(t,r'),(\hat{t},\hat{r})} \text{ with } B_{(t,r'),(\hat{t},\hat{r})} < B^{\max}. \tag{5.8}$$

Although the dominance criterion of Theorem 1 requires a strict subset, (5.8) includes timetable information such that strict route dominance is not guaranteed. The following example illustrates this. In the network of Figure 5.6, the two routes between A and B (r_1 and r_2) are indicated. The blocked link sets can be composed visually. First of all, route r_3 is the only route that is compatible with r_1. Thus all links that do not part r_3 belong to BL_{r_1}. Second, since no route is compatible with route r_2, the blocked link set of r_2 contains all links of the network. Thus $BL_{r_1} \subsetneq BL_{r_2}$. Together with the fact that r_1 and r_2 have the same origin and destination and use the same set of signals, Theorem 1 states that r_1 dominates r_2. In the this network, condition (5.8) with $r = r_1$

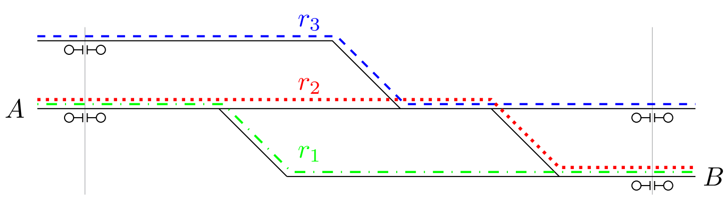


Figure 5.6: Counterexample of strict route dominance.

and $r' = r_2$ is only possible if $\hat{r} = r_3$. Since $r_1 \nleftrightarrow r_3$, $B_{(t,r_1),(\hat{t},r_3)} = \infty$ and (5.8) becomes $B_{(t,r_2),(\hat{t},r_3)} < B^{\max}$. However, there is no guarantee that there exists a train $\hat{t}$ that satisfies this criterion. Thus without timetable information, it cannot be assured that r_1 strictly dominates r_2.

Starting from the set of all possible routes (R), dominated routes can be pruned one by one. Doing so, one obtains the set of non-dominated routes. For simplicity, this set is also named R. In the remainder of this dissertation, it is assumed that R contains no dominated routes anymore.

5.1.2 Step 2: trainroute dominance

At the start of step 2, the set of non-dominated routes R is available. For each train in the (input) timetable ($t \in T$), its subset of correctly oriented routes connecting that train's inbound and outbound track with its platforms (R_t) can now be constructed. This gives the set of trainroutes $T \times R_t$. Since this set can be very large, a preprocessing process based on what is done in Zwaneveld et al. (2001) is applied. The main difference between Zwaneveld's approach and ours comes from the influence of the objective function which is different in both cases; Zwaneveld et al. (2001) maximize a weighted sum of the trainroutes, a linear function, while in (4.3), a weighted sum of pairs of trainroutes, which is a quadratic function, is minimized.

Where in step 1, routes are pruned based on the usage of resources, now, timetable information is used to reduce the number of candidate trainroutes. Similar to the dominance criterion of routes (5.5), it is said that trainroute (t, r) *(trainroute) dominates* another trainroute for the same train (t, r'), denoted by $(t, r) \preccurlyeq (t, r')$, if replacing (t, r') by (t, r) in any feasible solution has no negative impact on the objective function value of this solution. For trainroutes of different trains, we say that (t', r') is dominated by all trainroutes of train t if none of these trainroutes is compatible with (t', r'). In this case, (t', r') cannot be part of a feasible routing solution.

For trainroute (t, r), define its *dominance set* $Dom_{(t,r)}$ as the set of incompatible trainroutes of which the spreading cost with any other trainroute is not larger (not worse) than the spreading cost of (t, r) and that other trainroute.

$$Dom_{(t,r)} = \left\{ (t', r') \in T \times R_{t'} : \begin{array}{l} C_{(t,r),(t',r')} = 15 \text{ and} \\ \forall (\bar{t}, \bar{r}) \in T \times R_{\bar{t}} : C_{(t,r),(\bar{t},\bar{r})} \geq C_{(t',r'),(\bar{t},\bar{r})} \end{array} \right\}. \quad (5.9)$$

In terms of the minimum time spans, $Dom_{(t,r)}$ becomes

$$Dom_{(t,r)} = \left\{ (t',r') \in T \times R_{t'} : \forall (\bar{t},\bar{r}) \in T \times R_{\bar{t}} : \begin{array}{c} B_{(t,r),(t',r')} = 0 \text{ and} \\ B_{(t,r),(\bar{t},\bar{r})} \leq B_{(t',r'),(\bar{t},\bar{r})} \text{ or} \\ B_{(t',r'),(\bar{t},\bar{r})} \geq B^{\max} \end{array} \right\}. \tag{5.10}$$

In the following theorem, it is shown that trainroute (t, r) can be removed from $T \times R_t$, whenever $Dom_{(t,r)}$ is not empty.

Theorem 2. *If the set $Dom_{(t,r)} \neq \emptyset$ for trainroute $(t, r) \in T \times R_t$, then (t, r) is dominated by each element of its dominance set or (t, r) can never be part of a feasible routing solution.*

The proof of this theorem goes in two steps. Denote with (t', r') an element of $Dom_{(t,r)}$. First, the proof for $t = t'$ is given and then the situation where $t \neq t'$ is proven. Remember that when two trainroutes conflict, their spreading cost equals 15. The same holds for two trainroutes of the same train since only one can be selected.

Proof. Take an arbitrary $(t', r') \in Dom_{(t,r)}$. If $t = t'$ and $r, r' \in R_t$, then for any $(\bar{t}, \bar{r}) \in T \times R_{\bar{t}}$ that conflicts with (t, r'), the definition of $Dom_{(t,r)}$ indicates that $C_{(t,r),(\bar{t},\bar{r})} \geq C_{(t,r'),(\bar{t},\bar{r})} = 15$ such that (t, r) also conflicts with $(\bar{t}, \bar{r})$. Moreover, from (5.9), one sees that the costs in the spreading objective function (4.2) for any feasible solution using (t, r) will never be smaller than when using route r' instead of r for train t. This means that (t, r) is dominated by (t, r') since the latter always gives at least as good solutions.

If $t \neq t'$, then for each $\bar{r} \in R_{t'}$ holds that $C_{(t,r),(t',\bar{r})} \geq C_{(t',r'),(t',\bar{r})} = 15$. This means that (t, r) conflicts with all possible trainroutes for train t'. Thus (t, r) can never be part of a feasible solution. $\square$

An example illustrates the trainroute dominance. Consider the network and corresponding timetable of Figure 5.7 and Table 5.1. Six trains run through this network and only those coming from point B and going towards E have two route options. The last column in Table 5.1 shows the intervals during which each train reserves its trajectory through the network. Table 5.2 is the matrix with the minimum time spans between each pair of trainroutes. When there are no common resources, the time span equals ∞. $B^{\max}$ is used when the time span is at least $B^{\max}$ minutes such that the corresponding spreading costs becomes 0. The bottom line of Table 5.2 shows the dominance sets that

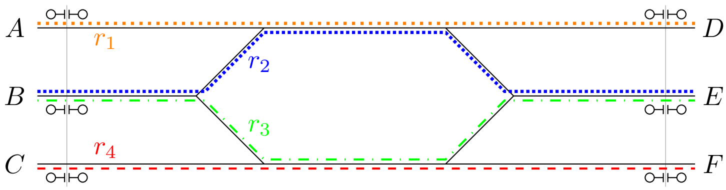


Figure 5.7: Example of trainroute dominance. The trains of Table 5.1 run through this network.

Table 5.1: Example of trainroute dominance: timetable of the trains running in the network of Figure 5.7.

train	**(from, to)**	**routeset**	**blocking time**
t_1	$(A,\ D)$	$R_{t_1} = \{r_1\}$	[0, 2]
t_2	$(B,\ E)$	$R_{t_2} = \{r_2, r_3\}$	[3, 5]
t_3	$(C,\ F)$	$R_{t_3} = \{r_4\}$	[20, 22]
t_4	$(B,\ E)$	$R_{t_4} = \{r_2, r_3\}$	[23, 25]
t_5	$(A,\ D)$	$R_{t_5} = \{r_1\}$	[40, 42]
t_6	$(B,\ E)$	$R_{t_6} = \{r_2, r_3\}$	[41, 43]

Table 5.2: Example of trainroute dominance: minimum time spans between each pair of trainroutes and the dominance sets for all trainroutes of the trains in Table 5.1.

	(t_1,r_1)	(t_2,r_2)	(t_2,r_3)	(t_3,r_4)	(t_4,r_2)	(t_4,r_3)	(t_5,r_1)	(t_6,r_2)	(t_6,r_3)
(t_1,r_1)	-	1	∞	∞	$B^{\max}$	∞	$B^{\max}$	$B^{\max}$	∞
(t_2,r_2)	1	-	0	∞	$B^{\max}$	$B^{\max}$	$B^{\max}$	$B^{\max}$	$B^{\max}$
(t_2,r_3)	∞	0	-	$B^{\max}$	$B^{\max}$	$B^{\max}$	∞	$B^{\max}$	$B^{\max}$
(t_3,r_4)	∞	∞	$B^{\max}$	-	∞	1	∞	∞	$B^{\max}$
(t_4,r_2)	$B^{\max}$	$B^{\max}$	$B^{\max}$	∞	-	0	$B^{\max}$	$B^{\max}$	$B^{\max}$
(t_4,r_3)	∞	$B^{\max}$	$B^{\max}$	1	0	-	∞	$B^{\max}$	$B^{\max}$
(t_5,r_1)	$B^{\max}$	$B^{\max}$	∞	∞	$B^{\max}$	∞	-	0	∞
(t_6,r_2)	$B^{\max}$	$B^{\max}$	$B^{\max}$	∞	$B^{\max}$	$B^{\max}$	0	-	0
(t_6,r_3)	∞	$B^{\max}$	$B^{\max}$	$B^{\max}$	$B^{\max}$	$B^{\max}$	∞	0	-
Dom	$\emptyset$	$\{(t_2,r_3)\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{(t_4,r_2)\}$	$\emptyset$	$\{(t_5,r_1),\ (t_6,r_3)\}$	$\emptyset$

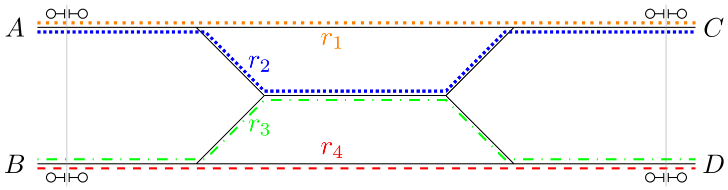


Figure 5.8: Illustration of the need for an iterative trainroute dominance removal procedure. This network is used by the trains of Table 5.3.

Table 5.3: Illustration of the need for an iterative trainroute dominance removal procedure: timetable of the trains running in the network of Figure 5.8.

train	**(from, to)**	**routeset**	**blocking time**
t_1	$(A,\ C)$	$R_{t_1} = \{r_1, r_2\}$	[0, 2]
t_2	$(B,\ D)$	$R_{t_2} = \{r_3, r_4\}$	[5, 7]

Table 5.4: Illustration of the need for an iterative trainroute dominance removal procedure: minimum time spans between each pair of trainroutes and dominance sets for all trainroutes of the trains in Table 5.3.

	(t_1, r_1)	(t_1, r_2)	(t_2, r_3)	(t_2, r_4)
(t_1, r_1)	-	0	∞	∞
(t_1, r_2)	0	-	3	∞
(t_2, r_3)	∞	3	-	0
(t_2, r_4)	∞	∞	0	-
Dom	$\emptyset$	$\{(t_1, r_1)\}$	$\{(t_2, r_4)\}$	$\emptyset$

are obtained using (5.10). Since their dominance set is not empty, trainroutes (t_2, r_2) and (t_4, r_3) are dominated by, respectively, (t_2, r_3) and (t_4, r_2). Next to the dominance by (t_6, r_3), trainroute (t_6, r_2) is dominated by the trainroute of t_5 because (t_6, r_2) is not compatible with any trainroute of t_5. As a result, trainroute (t_2, r_2) (or (t_4, r_3) or (t_6, r_2)) will not be used and can be removed, since the objective function (and the planning of the trains) is better when this trainroute is not used together with one of its dominators.

After the removal of a dominated trainroute, the dominance sets need to be updated (before other trainroutes are removed) since some new trainroutes can become dominated or non-dominated. The following example illustrates the need of an iterative removal procedure. According to the timetable of Table 5.3, two trains run through the network of Figure 5.8. The minimum time spans for each trainroute pair is shown in Table 5.4. Each of the trains has two possible routes, a non-dominated and a dominated one. Removing (t_1, r_2) from the set of trainroutes and updating the dominance sets, both (t_2, r_3) and (t_2, r_4) dominate each other, in which case they are called *equivalent*. In this case, only one can be pruned since otherwise no candidate trainroute for train t_2 remains.

This example shows that a set of trainroutes has no unique maximal subset of non-dominated trainroutes. However, similar to what is done by Zwaneveld et al. (2001), it can be shown that the number of non-dominated trainroutes does not depend on the order in which trainroutes are removed.

Theorem 3. *Given a set of trainroutes, the cardinality of each maximal subset of non-dominated trainroutes is independent of the dominance order.*

To proof this theorem, some other results are needed.

Lemma 1. *The trainroute dominance from Theorem 2 is transitive.*

Proof. Let $(t_1, r_1), (t_2, r_2)$ and (t_3, r_3) be three trainroutes for which holds that $(t_1, r_1) \preccurlyeq (t_2, r_2)$ and $(t_2, r_2) \preccurlyeq (t_3, r_3)$. According to (5.9), for each $(\bar{t}, \bar{r})$ holds that $C_{(t_1,r_1),(\bar{t},\bar{r})} \leq C_{(t_2,r_2),(\bar{t},\bar{r})} \leq C_{(t_3,r_3),(\bar{t},\bar{r})}$. Since this is valid for $(\bar{t}, \bar{r}) = (t_1, r_1)$ and $C_{(t_2,r_2),(t_1,r_1)} = 15$, one gets $C_{(t_3,r_3),(t_1,r_1)} = 15$. Thus $(t_1, r_1) \preccurlyeq (t_3, r_3)$. □

Lemma 2. *Given a set of trainroutes $X \subseteq T \times R$. Let $X' \subset X$ be obtained from X by removing, one by one, a number of dominated trainroutes. If trainroute (t, r) is dominated in X, then (t, r) is dominated in X' or (t, r) is equivalent to an element of $X \setminus X'$.*

In the proof of this lemma, only the case in which $X' = X \setminus \{(t', r')\}$ for a trainroute (t', r') is proven. For smaller subsets, the procedure of the proof can be repeated for one removed element at a time.

Proof. Consider four trainroutes $(t_i, r_i) \in X, i = 1, 2, 3, 4$, such that $(t_1, r_1) \preccurlyeq (t_2, r_2)$, $(t_3, r_3) \preccurlyeq (t_4, r_4)$, and $(t_2, r_2) \neq (t_4, r_4)$. Assuming that $X' = X \setminus \{(t_4, r_4)\}$, the dominance or equivalence of (t_2, r_2) needs to be proven.

For $(t_1, r_1) = (t_4, r_4)$, Lemma 1 gives $(t_3, r_3) \preccurlyeq (t_4, r_4) = (t_1, r_1) \preccurlyeq (t_2, r_2)$ such that $(t_3, r_3) \preccurlyeq (t_2, r_2)$. Thus (t_2, r_2) is still dominated in X' or, in case $(t_2, r_2) = (t_3, r_3)$, it is equivalent to (t_4, r_4). If $(t_1, r_1) \neq (t_4, r_4)$, nothing changes with respect to (5.9) and thus $(t_1, r_1) \preccurlyeq (t_2, r_2)$. □

Lemma 2 gives rise to the following corollary.

Corollary 1. *If two dominated trainroutes (t, r) and (t', r') are equivalent, then, except for (t, r) and (t', r'), the sets of dominated trainroutes after the removal of (t, r) or (t', r') are equal.*

This corollary implies that replacing a trainroute by another, equivalent trainroute does not affect the sets of dominated and non-dominated trainroutes. Using these results, the proof of Theorem 3 can be formulated. The principle of induction on the number of removed trainroutes is used.

Proof of Theorem 3. Let A, B be two maximal subsets of non-dominated trainroutes of a set of trainroutes X such that $A = X \setminus \{a_1, \ldots, a_n\}$ and $B = X \setminus \{b_1, \ldots, b_m\}$, where the trainroutes a_i and b_j are removed in the order indicated by their indices. For the ease of notation, A_i is used to indicate the set $X \setminus \{a_1, \ldots, a_i\}$ for $i = 1, \ldots, n$ and similar for B_j with $j = 1, \ldots, m$.

Consider the case where $n = 1$. Since b_1 is dominated in X and A contains no dominated trainroutes, Lemma 2 states that a_1 and b_1 are equivalent. If $m > 1$, then b_2 must be dominated in B_1 and, according to Corollary 1, also in A. This contradicts with the definition of A and thus $m = 1 = n$.

Now follows the induction step. Assume Theorem 3 holds for $n - 1$, with $n > 1$. If a_1 and b_1 are equivalent, the theorem holds for A_1 and B_1 and thus Corollary 1 proves this case. However, if a_1 and b_1 are not equivalent, the ordered set $\{b_1, \ldots, b_m\}$ can be transformed such that a_1 becomes the first element of this transformed set what finishes the proof.

In case $a_1 \notin \{b_1, \ldots, b_m\}$, the definition of B as non-dominated subset of X and Lemma 2 for a_1 give that there exists an index $j \in \{1, \ldots, m\}$ such that a_1 is equivalent to b_j. Let j^* be the smallest index such that b_{j^*} is equivalent with a_1. Using Corollary 1, b_{j^*} can be replaced by a_1 without changing the set B (except for equivalent trainroutes). If $a_1 \in \{b_1, \ldots, b_m\}$, say $a_1 = b_{j'}$, then the positions of b_{j^*} and $b_{j'}$ should be switched. From the selection of j^*, the fact that a_1 is dominated in X, and Lemma 2 follows that a_1 is dominated in all sets B_k with $k = 1, \ldots, j^* - 1$. Since this holds for $k = j^* - 2$, Lemma 2 states that b_{j^*-1} is dominated in $B_{j^*-2} \setminus \{a_1\}$ or b_{j^*-1} is equivalent with a_1 but this contradicts the definition of j^*. Thus the order of $b_{j^*} = a_1$ and b_{j^*-1} in $\{b_1, \ldots, b_m\}$ can be switched without affecting B. This procedure can be continued until a_1 becomes the first element of $\{b_1, \ldots, b_m\}$. □

The usage of timetable information makes the trainroute dominance stronger than the route dominance. Nevertheless, it is easy to see that any trainroute with a dominated route is trainroute dominated. The advantage of using both dominance rules is that the initial number of trainroutes and the necessary amount of computations is reduced. Similar to the strict route dominance, one can introduce the notion of strict trainroute dominance. The example of Figure 5.8 and Tables 5.3 and 5.4, in which two trainroutes dominate each other, illustrates that Theorem 2 does not imply strict dominance.

5.1.3 Step 3: mathematical model for the TRP

The last step of the routing module is to solve the TRP on the set of non-dominated trainroutes. Therefore, the TRP is modeled as a node packing problem (NPP). Define for all trainroutes $(t, r) \in T \times R_t$ the binary decision variable

$$x_{(t,r)} = \begin{cases} 1 & \text{if route } r \text{ is selected for train } t, \\ 0 & \text{otherwise.} \end{cases}$$

Using these decision variables, the mathematical model for the TRP can be constructed.

$$\text{Minimize} \sum_{t \in T} \sum_{t' \in T} \sum_{r \in R_t} \sum_{r' \in R_{t'}} C_{(t,r),(t',r')} \cdot x_{(t,r)} \cdot x_{(t',r')} \tag{5.11}$$

subject to

$$x_{(t,r)} + x_{(t',r')} \leq 1 \quad \forall (t,r) \in T \times R_t, (t', r') \in T \times R_{t'} : B_{(t,r),(t',r')} = 0, \tag{5.12}$$

$$\sum_{r \in R_t} x_{(t,r)} = 1 \quad \forall t \in T, \tag{5.13}$$

$$x_{(t,r)} \in \{0, 1\} \quad \forall (t,r) \in T \times R_t. \tag{5.14}$$

The quadratic objective function (5.11) minimizes the sum of all the spreading costs (4.2). For each conflicting couple of trainroutes, (5.12) implies that only one of these trainroutes can be selected. Constraint (5.13) ensures that each train gets exactly one route assigned and (5.14) bounds the variables.

In general, a solver is able to find optimal solutions using the model (5.11)-(5.14) for limited instances only. In order to make it suitable for the purposes of this dissertation, some improvements are required. At first, the inequalities of the form (5.12) can be strengthened. Similar to the generation of clique inequalities in Zwaneveld (1997) and Zwaneveld et al. (2001), (5.12) is replaced by the stronger clique inequalities

$$x_{(t,r)} + \sum_{r' \in R^c_{t'}} x_{(t',r')} \leq 1 \qquad \begin{array}{l} \forall (t,r) \in T \times R_t, t' \in T \setminus \{t\}, \\ R^c_{t'} = \{r' \in R_{t'} : B_{(t,r),(t',r')} = 0\}. \end{array} \tag{5.15}$$

The reasoning behind these clique inequalities is as follows. Consider three mutually conflicting trainroutes $(t_1, r_1), (t_2, r_2), (t_3, r_3)$. Summing $x_{(t_i,r_i)} + x_{(t_j,r_j)} \leq 1$ for all combinations of $i, j = 1, 2, 3$, gives $2x_{(t_1,r_1)} + 2x_{(t_2,r_2)} + 2x_{(t_3,r_3)} \leq 3$. Since the left hand side is even, the right hand side cannot be odd and can be lowered by 1. Dividing by 2 then gives $x_{(t_1,r_1)} + x_{(t_2,r_2)} + x_{(t_3,r_3)} \leq 1$ which is exactly what is done in (5.15).

A second improvement to the model is to linearize the quadratic objective function. Kroon et al. (2008c) face the same problem and compare some linearisation methods. The technique of Lawler (1963) is compared with the one of Adams et al. (1994) and Kaufman et al. (1978). Since our cost structure (4.2) depends on trainroutes and not only on trains, the so-called *special Lawler formulation* that is used in Kroon et al. (2008c), is not applicable to our problem. In the following, the three other linearisation methods are compared. For the ease of notation, each of the models is named after the (first) author of the corresponding article.

Lawler-TRP model

The oldest linearisation technique that is considered, comes from Lawler (1963) and uses for each pair of trainroutes an extra decision variable y which is continuous in the interval between 0 and 1.

$$0 \leq y_{(t,r),(t',r')} \leq 1 \quad \forall (t,r) \in T \times R_t, (t',r') \in T \times R_{t'}. \tag{5.16}$$

This set of variables is introduced to replace the quadratic term $x_{(t,r)} \cdot x_{(t',r')}$ of (5.11). Thus, $y_{(t,r),(t',r')}$ should be 1 if both $x_{(t,r)}$ and $x_{(t',r')}$ equal 1. In order to ensure this, extra constraints are added.

$$x_{(t,r)} + x_{(t',r')} \leq 1 + y_{(t,r),(t',r')} \quad \forall (t,r) \in T \times R_t, (t',r') \in T \times R_{t'}. \tag{5.17}$$

Together with the linearized objective function

$$\text{minimize} \sum_{t\in T}\sum_{t'\in T}\sum_{r\in R_t}\sum_{r'\in R_{t'}} C_{(t,r),(t',r')} \cdot y_{(t,r),(t',r')}, \tag{5.18}$$

constraint set (5.17) provides a lower bound on the continuous decision variables y. Since the objective is to minimize a nonnegative cost times y, there is an incentive to set $y_{(t,r),(t',r')} = 0$. When both $x_{(t,r)}$ and $x_{(t',r')}$ equal 1, constraint set (5.17) forces $y_{(t,r),(t',r')}$ to be larger than or equal to 1. Thanks to (5.17), the variables y are symmetric with respect to the indices: $y_{(t,r),(t',r')} = y_{(t',r'),(t,r)}$. The Lawler-TRP model (5.18), (5.13)-(5.17) is used in Dewilde et al. (2014).

Adams-TRP model

The same set of variables is used by Adams et al. (1994). The objective function (5.18) is not changed, but instead of using (5.17), the following constraints are added to the model.

$$x_{(t,r)} = \sum_{r'\in R_{t'}} y_{(t,r),(t',r')} \quad \forall (t,r)\in T\times R_t, t'\in T \tag{5.19}$$

and

$$x_{(t,r)} = \sum_{r'\in R_{t'}} y_{(t',r'),(t,r)} \quad \forall (t,r)\in T\times R_t, t'\in T. \tag{5.20}$$

For $x_{(t,r)} = 0$, the set of constraint (5.19) implies that $y_{(t,r),(t',r')} = 0$ for all (t', r'), and (5.20) ensures that $y_{(t',r'),(t,r)} = 0$ for all (t', r'). As a consequence, only these combinations for which both $x_{(t,r)}$ and $x_{(t',r')}$ are 1 yield a strict positive $y_{(t,r),(t',r')}$ and thus $y_{(t,r),(t',r')} = x_{(t,r)} \cdot x_{(t',r')}$.

Kaufman-TRP model

The linearisation technique of Kaufman et al. (1978) uses other decision variables but the linkage with the initial decision variables x is similar to (5.17). Define for all trainroutes $(t,r) \in T \times R_t$ the non-negative, continuous variable $z_{(t,r)}$,

$$z_{(t,r)} \geq 0 \quad \forall (t,r)\in T\times R_t. \tag{5.21}$$

By adding the constraints

$$\sum_{t'\in T}\sum_{r'\in R_{t'}} C_{(t,r),(t',r')} \left(x_{(t,r)} + x_{(t',r')} - 1\right) \leq z_{(t,r)} \quad \forall (t,r)\in T\times R_t, \tag{5.22}$$

the objective function can be replaced by

$$\text{minimize} \sum_{t \in T} \sum_{r \in R_t} z_{(t,r)}. \tag{5.23}$$

The summands in the left hand side of (5.22) can only be strictly positive if $x_{(t,r)} = x_{(t',r')} = 1$. In this case $C_{(t,r),(t',r')}$ is added to the lower bound for $z_{(t,r)}$. In the end, equality will arise due to (5.23).

In comparison with the other linearisation techniques, the dimension of the z variables is smaller than the dimension of the y variables. Next to that, the three formulations differ in size of the resulting model, the quality of the LP relaxation, and the time required to solve the problem. Table 5.5 compares the three formulations. The first two rows contain the references to the objective function and the necessary constraints. Regarding the number of variables that are needed for the linearisation, the variables y have four indices such that there are $\mathcal{O}\big(|T|^2|R|^2\big)$ variables, while z has only two indices. A similar reasoning applies for the constraints.

Table 5.5: Comparison of the linearisation methods. The first two rows contain the references to the formulas that summarize the MILP models. The entries in the rows ♯*variables* and ♯*constraints* correspond to the number of variables and constraints that are due to the linearisation technique. The following two rows present the *LP gap* and corresponding solving time (*cpu*) of the models during the first two iterations of the algorithm for the NSC case study. At the bottom, the average computation time to solve each model in the course of the algorithm and its worst case value are given.

	Lawler-TRP	**Adams-TRP**	**Kaufman-TRP**
objective function	(5.18)	(5.18)	(5.23)
constraints	(5.13)-(5.17)	(5.13)-(5.16), (5.19)-(5.20)	(5.13)-(5.15), (5.21)-(5.22)
♯variables	$\mathcal{O}\big(\|T\|^2\|R\|^2\big)$	$\mathcal{O}\big(\|T\|^2\|R\|^2\big)$	$\mathcal{O}(\|T\|\|R\|)$
♯constraints	$\mathcal{O}\big(\|T\|^2\|R\|^2\big)$	$\mathcal{O}\big(\|T\|^2\|R\|\big)$	$\mathcal{O}(\|T\|\|R\|)$
LP gap^1 (cpu)	10.59% (0.39 s)	16.56% (0.73 s)	6.53% (0.25 s)
LP gap^2 (cpu)	65.37% (1.12 s)	60.72% (3.74 s)	30.92% (0.38 s)
average cpu	1.02 s	2.94 s	0.55 s
worst case cpu	1.20 s	3.82 s	0.63 s

The next two rows in Table 5.5 represent the *LP gap* and computation time (cpu) of solving each mathematical model for the first two iterations of the algorithm for the NSC case study[10]. In Kroon et al. (2008c), it is shown that for their settings[11] the LP relaxation of the Kaufman-TRP model will never be better than when the Lawler-TRP formulation is used. Furthermore, the linearisation technique of Adams et al. (1994) is supposed to give the best LP relaxation of the three. However, considering the results in Table 5.5, one sees that these claims do not hold for our problem. Thanks to the smaller problem size of the Kaufman-TRP formulation and the internal preprocessing by the solver itself (not to be confused with the preprocessing of step 1 and step 2) smaller LP gaps arise and shorter computation times are needed when solving the Kaufman-TRP model. In both iterations, the Kaufman-TRP model performed best. Although these results are only for the first two iterations of the algorithm and for one particular case study, similar results are obtained for all tests that are made. The last two rows of Table 5.5 contain the average and worst case computation time the solver needs for each passage of the routing module during the full algorithm on the NSC case study. Also in these rows, the results for the Kaufman-TRP formulation are better than when the other two linearisation techniques are used; the average solver time is up to 1.8 and 5.3 times shorter. Moreover, for the larger instances of Chapter 8, the differences become even more apparent with the average computation times to solve the Kaufman-TRP model being up to 6.3 times shorter than for the Lawler-TRP model and 16 times shorter than when the Adams-TRP formulation is used.

Based on this comparison, the linearisation technique of Kaufman et al. (1978) is used for the MILP model to solve the TRP in our algorithm. The objective function is (5.23) and (5.13)-(5.15) and (5.21)-(5.22) are used as constraints.

[10] As explained in Section 4.1, the developed algorithm works iteratively. Thus, in the course of the algorithm, the routing module is called several times. For the results in these rows, only the first two calls are considered.

[11] In Kroon et al. (2008c), the platform assignment is seen as part of the TRP. Since we separate the two, there is a difference in settings what makes the conclusions of Kroon et al. (2008c) not necessarily valid for our approach.

The **Kaufman-TRP model** is the following.

$$\text{Minimize} \sum_{t \in T} \sum_{r \in R_t} z_{(t,r)}.$$

Subject to

$$\sum_{t' \in T} \sum_{r' \in R_{t'}} C_{(t,r),(t',r')} \big(x_{(t,r)} + x_{(t',r')} - 1 \big) \leq z_{(t,r)} \qquad \forall (t,r) \in T \times R_t,$$

$$\sum_{r \in R_t} x_{(t,r)} = 1 \qquad \forall t \in T,$$

$$x_{(t,r)} + \sum_{r' \in R^c_{t'}} x_{(t',r')} \leq 1 \qquad \forall (t,r) \in T \times R_t, t' \in T \setminus \{t\}, \quad R^c_{t'} = \{ r' \in R_{t'} : B_{(t,r),(t',r')} = 0 \},$$

$$z_{(t,r)} \geq 0 \qquad \forall (t,r) \in T \times R_t,$$

$$x_{(t,r)} \in \{0,1\} \qquad \forall (t,r) \in T \times R_t.$$

5.2 Computational results

5.2.1 Framework

As can be seen in the overview Algorithm 5.2, solving the TRP occurs at the beginning of the algorithm or after a successful call to the timetabling module or the platforming module. The first time the routing module is applied, a routing solution for the reference timetable is computed. Afterwards, the currently active timetable is used as input. During the TRP, no changes are made to the timetable and it is assumed that the assigned platforms in each station are fixed. As a consequence, only the routes through the grids between the outer stations and the tunnel are flexible. In the following section, the impact of the preprocessing steps and of the optimized routes through the grids are analyzed.

5.2.2 Results of the routing module

Often, preprocessing is performed to reduce the problem size and required computation time without losing solution quality. This is also done in this dissertation with the two dominance rules of Sections 5.1.1 and 5.1.2. To discuss the impact of the route dominance and trainroute dominance, the total

Algorithm 5.2 Framework of the developed algorithm: the routing module

input: infrastructure data and reference timetable
while number of consecutive non-improving iterations $\leq iter^{\max}$ **do**
 solve the train routing problem (TRP)
 apply tabu search (timetabling module)
 if timetabling did not yield improvements
  then start the platforming module

Table 5.6: Impact of route dominance (*Rdom*) and trainroute dominance (*TRdom*). The impact of using these dominance rules on the total number of (train-)routes and on the performance of the solver is summarized. The performance of the solver is measured by the *LP gap* of each instance and the required time to obtain the optimal solution (*cpu*).

Rdom	# routes	# northbound	TRdom	# trainroutes	LP gap (%)	cpu (s)
N	12252	6219	N	1874	97.63%	35.85 s
N	12252	6219	Y	85	6.53%	0.27 s
Y	2738	1404	N	411	68.16%	3.21 s
Y	2738	1404	Y	85	6.53%	0.25 s

number of candidate routes and trainroutes are compared before and after the preprocessing. This is done in Table 5.6. In this table, *Rdom* refers to route dominance and *TRdom* to trainroute dominance. The entries *Y* (yes) and *N* (no) indicate whether or not the corresponding dominance phase is applied or not. Initially, there are 12252 routes through the network of the NSC like it is depicted in Figure 4.4. From these 12252, 6219 routes are northbound, the other 6033 routes are directed towards station Midi. Note that, without the predetermined platform orientations, the total number of (unoriented) routes rises to 30911. After the preprocessing based on the blocked link sets, 2738 routes remain. This is a reduction of more than 75%.

In case no preprocessing is applied, there are 1874 trainroutes for the timetable with 80 trains. Thanks to the route dominance, this set reduces to 411. Applying the trainroute dominance of Section 5.1.2, the set of non-dominated trainroutes contains only 85 elements. This corresponds to an extra reduction of nearly 80%. In total, more than 95% of all trainroutes are removed from the initial set of candidate trainroutes. Since the route dominance is a special case of trainroute dominance, the usage of the route dominance rule has no influence on the final number of non-dominated trainroutes. However, more computation steps are required during this preprocessing step, for example, to compute the dominance sets (5.9). Therefore, there is a small difference in computation time to solve the entire TRP. This difference becomes larger when the size of the problem grows. Since the route dominance is performed before the entire algorithm starts and only needs to be done once, the required time for this step is ignored.

From the results in Table 5.6, it becomes clear that the preprocessing is advantageous for the solver since the LP gap for the smaller instances is considerably smaller. As a consequence, the solver times are much shorter. The entire algorithm is coded in C++ (Visual Studio 2010) on a 2009 DELL Optiplex 760 with Intel(R) Core(TM) 2 Duo 3.00 GHz, 4.00 GB RAM, 64-bit operating system. Cplex 12.5 is used to solve the mathematical models.

5.2.3 Simulation results

When the Kaufman-TRP model is solved, the impact on the robustness and other performance indicators of the optimized routing can be simulated. Therefore, the simulation model of Section 3.3.4 is used. The reference system with the input timetable and routing is used for comparison. The simulation output is summarized in Tables 5.7-5.10. The difference between these tables comes from the delay scenarios that are used. According to the notation introduced in Section 3.3.4, the delay scenario of Table 5.7, $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$, should be interpreted as follows. The first entry, $E_{|T|/2}$, indicates that half of the trains are delayed upon arrival at the network's boundary and the size of these delays is drawn from the exponential distribution with each train's average arrival delay as parameter. Next to delays upon arrival, half of the trains face a dwell delay of 30 seconds during their stop at the Central station $\left(P^{(0.5)}_{|T|/2}\right)$. No external delays are added to the trains in the two outer stations.

One value is equal in all tables, this is the improvement in *spreading cost*, the objective of the algorithm. Since this value is computed within the algorithm and not in the simulation module, there is no influence of delays. For this criterion, a reduction from 100 to 76.3% is achieved by optimizing the routes of the trains through the network.

The reduced value for Rob_1 and the increased value for Rob_2, which are computed using (3.4) and (3.5), indicate that the robustness of the optimal routing solution is slightly better than that of the reference system. The standard deviations of the RWTT (Rob_{stdev}), the average delays per passenger (*pax delays*) and the average delays of all trains (*train delays*) are, statistically seen, equal for both systems. For the total amount of *knock-on* delays, a significant reduction in the size of the standard deviation is detected in all delay scenarios. Independent of whether the standard deviations are equal or not, the reduction in delays for

Table 5.7: Simulation results of the optimal routing solution for delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. Next to the spreading cost, the two robustness scores of Section 3.3.3 and the robustness' standard deviation (Rob_{stdev}) are given in the first rows. Then, for the arrival delay per passenger (*pax delays*), the total amount of *train delays*, and the total amount of propagated delays (*knock-on*), the average values and the standard deviations (between brackets) are given. The last three rows contain the percentages of *newly*, respectively, *extra delayed* trains as defined in Section 3.3.4, and the maximal amount of train delays that were found during any iteration of the simulation run (*worst case*).

	reference	routing
spreading cost (%)	100	76.3
Rob_1 (%)	100	99.5
Rob_2 (%)	100	101.1
Rob_{stdev} (min)	0.861	0.855
pax delays (min)	1.69 (0.287)	1.67 (0.285)
train delays (min)	162 (23.4)	159 (23.2)
knock-on (min)	35.2 (10.1)	32.5 (9.8)
newly delayed (%)	8.42 (2.85)	7.54 (2.75)
extra delayed (%)	33.5 (5.68)	30.0 (5.52)
worst case (min)	298	289

Table 5.8: Simulation results of the optimal routing solution for delay scenario $\left(E_{|T|/2}, 0, E_{|T|/2}, 0\right)$.

	reference	routing
spreading cost (%)	100	76.3
Rob_1 (%)	100	99.5
Rob_2 (%)	100	101.1
Rob_{stdev} (min)	0.878	0.872
pax delays (min)	1.70 (0.293)	1.68 (0.291)
train delays (min)	164 (24.2)	161 (24.2)
knock-on (min)	37.1 (10.7)	34.5 (10.4)
newly delayed (%)	10.18 (3.12)	9.19 (3.02)
extra delayed (%)	34.6 (5.56)	31.2 (5.44)
worst case (min)	298	293

Table 5.9: Simulation results of the optimal routing solution for delay scenario $\left(E_{3|T|/4}, 0, P^{(0.5)}_{|T|/2}, 0\right)$.

	reference	routing
spreading cost (%)	100	76.3
Rob_1 (%)	100	99.5
Rob_2 (%)	100	100.9
Rob_{stdev} (min)	1.001	1.000
pax delays (min)	2.39 (0.334)	2.37 (0.333)
train delays (min)	222 (26.8)	220 (26.7)
knock-on (min)	42.2 (10.4)	39.3 (10.2)
newly delayed (%)	5.69 (2.28)	5.20 (2.25)
extra delayed (%)	38.4 (5.43)	34.8 (5.29)
worst case (min)	369	360

Table 5.10: Simulation results of the optimal routing solution for delay scenario $\left(P^{(0.5\hat{D})}_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$.

	reference	routing
spreading cost (%)	100	76.3
Rob_1 (%)	100	99.7
Rob_2 (%)	100	100.8
Rob_{stdev} (min)	0.168	0.169
pax delays (min)	1.28 (0.056)	1.27 (0.056)
train delays (min)	121 (4.31)	119 (4.29)
knock-on (min)	14.7 (2.64)	13.1 (2.59)
newly delayed (%)	6.87 (2.36)	6.16 (2.30)
extra delayed (%)	23.5 (3.24)	20.9 (3.21)
worst case (min)	135	133

each of these performance indicators is found to be significant. The last rows in Tables 5.7-5.10 contain the percentages of *newly delayed* and *extra delayed* trains and also the maximal amount of train delays that have arisen during one of the iterations of the simulation module (*worst case*). For the percentages, the standard deviation as well as the average values are significantly smaller for the optimal routing than for the reference system.

The importance of investigating the robustness of routing through the grids

Although the improvements based on only the routing module are statistically significant, we think these improvements are not significant enough for the passengers. Together with the timetabling and platforming modules of the next chapters, the size of these improvements will further increase. Moreover, the fact that the results in Tables 5.7-5.10 are not that spectacular is not unexpected since the only change that is made in comparison with the reference situation is the routing of the trains through the grids. However, if one takes a closer look at the amount of delay propagation within the grids, like is done in Table 5.11, the real impact of the routing module can be seen. The location of the grids are indicated in Figure 5.9. Since trains mainly run straight on in *grid* CM_1, the grids between stations Central and Midi are grouped and considered as one (*grid* CM). Where in the stations, the amount of knock-on delays is more or less stable, a reduction in the amount of propagated delays of 23% to 28% ($= 1 - 3.4/4.7$) is measured on the grids. This endorses the quotation of the beginning of this chapter that investigating the routing of trains through station areas is highly relevant for improving the punctuality of a railway system.

Another argument to illustrate the importance of investigating the routing of trains is found by comparing the impact of a conflict within a grid (*grid conflict*) or at a station (*station conflict*). The limited capacity within the NSC, or more specifically the tunnel with the Central station, is often seen as the weakest link of the network. Nevertheless, the crisscross of routes at the grids makes that the set of hindered trains often is larger due to a conflict in one of the grids than due to an extended stop. The results of Table 5.12 confirm this. This table has the same layout as Tables 5.7-5.10. Instead of any of the delay scenarios of these tables, a single conflict of predetermined size is triggered at the Central station (column *Central*) or at one of the grids such as indicated in Figure 5.9 (the other columns). For the results in this table, the initial disturbance that is inserted in the system equals 5 minutes. This value is chosen because of the clear differences it gives. For other (smaller) disturbances, the trend is similar.

Table 5.11: The amount of propagated delays (in minutes) in the stations and on the grids of the optimal routing solution for delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. The locations of the grids are indicated in Figure 5.9.

	North	grid NC	Central	grid CM	Midi
reference	7.3	4.7	16.0	4.1	3.0
routing	7.2	3.4	15.9	3.1	3.0

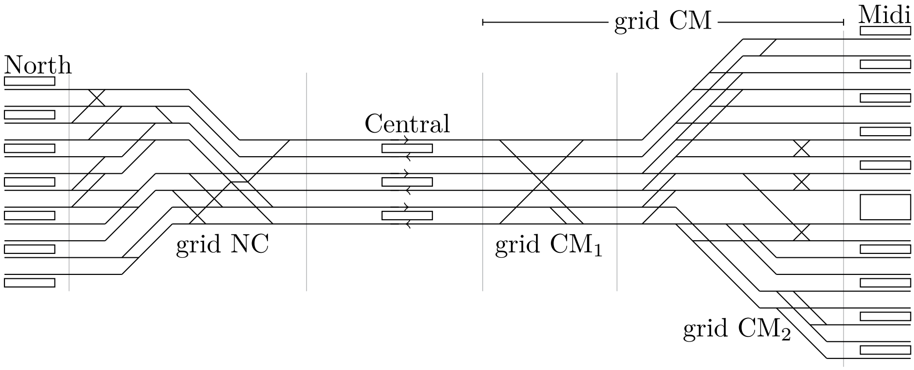


Figure 5.9: The grids within the NSC.

Compared with the impact of one delayed train in the Central station, the worsening in RWTT and RWTT_{ext} due to a grid conflict is larger and more (knock-on) delays arise. What is remarkable, is that the total amount of passengers' delays[12] is considerably higher when the disturbance happens in one of the grids. This comes from the fact that the total number of passengers that are in the train at the moment of the disturbance or will board the disturbed train, is, on average, larger for the grid conflicts than for the conflicts in the Central station. The other numbers in the table show that grid conflicts cause more harmed trains what also leads to considerably more delays in the worst case scenario. A delay in grid CM_1 has a smaller effect on the system than in grid NC or grid CM_2. This is because the blocking times of the sector that corresponds to grid CM_1 is smaller than that of the other grids and there are hardly trains that make use of these switches. This also holds for the routing solution. Doing the same exercise as in Table 5.12 for the system with the optimal routing, it can be seen in Table 5.13, that the impact of the grid conflicts reduces. This is expected since the routing module improves the spreading on the grids.

[12] To get relevant numbers, the amount of passengers' delays is summed over all passengers instead of averaged like in the previous tables.

Table 5.12: Comparison of the impact of a conflict within a grid or at the Central station for the *reference system*. Unlike in the previous tables, the passengers' delays are summed over all passengers to get meaningful numbers.

	Central	grid NC	grid CM_1	grid CM_2
Rob_1 (%)	100	100.6	100.3	100.8
Rob_2 (%)	100	82.5	90.5	76.3
Rob_{stdev} (min)	0.093	0.130	0.128	0.151
pax delays (min)	1408 (689)	1654 (972)	1541 (955)	1742 (1123)
train delays (min)	9.7 (3.3)	10.5 (4.0)	9.6 (4.3)	10.8 (5.0)
knock-on (min)	4.7 (3.3)	5.5 (4.0)	4.6 (4.3)	5.8 (5.0)
newly delayed (%)	2.51 (1.78)	3.06 (2.13)	2.71 (2.29)	3.40 (2.64)
extra delayed (%)	2.60 (1.84)	3.09 (2.13)	2.77 (2.33)	3.47 (2.68)
worst case (min)	18.7	23.8	28.2	33.5

Table 5.13: Comparison of the impact of a conflict within a grid or at the Central station for the *optimal routing solution*.

	Central	grid NC	grid CM_1	grid CM_2
Rob_1 (%)	100	100.2	100.2	100.4
Rob_2 (%)	100	92.7	94.5	86.0
Rob_{stdev} (min)	0.091	0.125	0.116	0.125
pax delays (min)	1392 (677)	1494 (928)	1468 (861)	1586 (928)
train delays (min)	9.6 (3.3)	9.5 (3.4)	9.1 (3.4)	9.8 (3.8)
knock-on (min)	4.6 (3.3)	4.5 (3.4)	4.1 (3.4)	4.8 (3.8)
newly delayed (%)	2.39 (1.64)	2.46 (1.75)	2.33 (1.78)	2.80 (2.04)
extra delayed (%)	2.43 (1.67)	2.48 (1.76)	2.36 (1.81)	2.83 (2.05)
worst case (min)	17.4	20.3	18.8	19.0

5.3 Comparison with other approaches

The presented method to solve the TRP is related to some other solution approaches that are found in literature. Although feasibility was the initial objective of the approach of Zwaneveld (see Section 2.4.1), the formulation of the model as an NPP and the idea of trainroute dominance proved to be useful when designing the routing module.

In Kroon et al. (2008c), the robustness of the routing solution is considered. Therefore, the model of Zwaneveld is extended with an objective function that is similar to ours. The main difference is twofold. First, the cost structure they use does not depend on the buffer times in the critical block section, but on a general cost that is determined by the trains and not by the routes. Second, in their problem, the platform assignment is considered flexible and thus more degrees of freedom arise than when the platform assignment is predefined like in our approach. Instead of using preprocessing, Kroon et al. (2008c) opt for aggregation. The computational results report that less than one hour is required to solve their instance with 55 trains and 88 aggregated routes. In our approach, the preprocessing in step 1 and step 2 are used instead of aggregation to reduce the set of routes. As a consequence, the Kaufman-TRP model finds its solutions in 0.63 seconds (worst case value in Table 5.5) for the NSC case study with 80 trains. Nevertheless, due to the difference in settings, it is hard to compare the performance of both approaches.

Instead of minimizing the total spreading cost, the *approach of Caimi* (Caimi et al. 2005) is to maximize the α smallest minimum time spans of the entire system. To compare their approach with ours, the same assumptions as in Section 4.4 are used and the Kaufman-TRP model is adapted to the objective function of the model of Caimi. The results of the comparison are summarized in Table 5.14. In this table, not only the original value of $\alpha = 4$ (Caimi et al. 2005) is used, but different values of α are compared. Although the preprocessing rules from step 1 and step 2 are used and the setup of the adapted model does not require significantly more computational effort, the computation time the solver needs to find the optimal solution (*cpu*) increases. The computation times and high interaction rate of their case study are the reasons why Caimi et al. (2005) opt for a heuristic solution procedure to solve the TRP.

Independent of whether the objective of Caimi or the spreading objective function is used, the α smallest minimum time spans of the optimal solutions that are returned by the solver are equal. This is not surprising since the spreading costs of (4.2) are inversely related to the size of the minimum time spans such that the smallest time spans have the largest costs. For the other, larger time spans, differences arise because these time spans are not considered in the approach of Caimi but are counted in the spreading objective function. In case of differences, the time spans of the Kaufman-TRP model are larger (better) than those of the Caimi model. The size of the smallest difference in minimum time span is added in the column *smallest difference* of Table 5.14. For example, for *Caimi (4)*, there is one minimum time span of 72 seconds extra compared to the optimal spreading solution that has an extra time span of 144 seconds instead. As a consequence, the *spreading costs* of the Caimi solutions increase. Note that for $\alpha = 8$ or 10, the solution with optimal spreading cost is returned. However, for $\alpha = 9$ this is not the case. Due to the optimization of the smallest minimum time spans only, the propagation of delays will, in general, be larger using the Caimi approach. This is confirmed by the simulation results that lie in between these of the reference system and the optimal routing. Thus, one can conclude that the spreading objective function outperforms the objective of Caimi et al. (2005) for the NSC case study.

Table 5.14: Comparison of the performance of the Kaufman-TRP model with the approach of Caimi et al. (2005). The selected values for α are added between parentheses. In the original paper of Caimi et al. (2005), α equals 4. This table contains the required computation time to solve each model (*cpu*), the size of the smallest minimum time span that is worse in the solution of Caimi et al. (2005) than in the Kaufman-TRP solution (*smallest difference*), and the resulting *spreading cost*.

	cpu (s)	smallest difference	spreading cost
Kaufman-TRP	0.25 s	-	100%
Caimi ($\alpha = 1$)	0.77 s	0 s	109.81%
Caimi ($\alpha = 2$)	2.84 s	0 s	117.15%
Caimi ($\alpha = 3$)	16.90 s	12 s	107.59%
Caimi ($\alpha = 4$)	22.98 s	72 s	100.25%
Caimi ($\alpha = 5$)	33.98 s	72 s	100.79%
Caimi ($\alpha = 6$)	39.09 s	72 s	100.79%
Caimi ($\alpha = 7$)	44.59 s	72 s	100.79%
Caimi ($\alpha = 8$)	55.24 s	-	100%
Caimi ($\alpha = 9$)	60.31 s	72 s	100.25%
Caimi ($\alpha = 10$)	86.70 s	-	100%

5.4 Conclusion

In this chapter, the applied solution technique for solving the TRP is explained. Since station areas often are the main source of (knock-on) delays, improving the robustness of the train routing is highly valuable. The presented methodology is composed of three steps. The first two steps consist of a preprocessing phase based on the blocked link dominance of routes and a dominance rule to prune trainroutes with a non-empty dominance set. For both preprocessing techniques, examples are given to illustrate its usage, some properties are proven, and its efficiency is shown. The third step is the construction of a mathematical model to solve the TRP for the set of non-dominated trainroutes. After discussing and comparing some improvements to this MILP model, the computational results are studied. The necessary computation times are analyzed and the improvement in spreading cost and delay propagation on the grids, where the routing acts on, are found remarkable. By improving the routing through the station areas, the impact of a single disturbance in one of the grids can be reduced. Nevertheless, this impact remains larger than a single dwell delay at the Central station. Finally, the presented approach is compared with other, related methodologies. For the purpose of this dissertation, the developed methodology is shown to be more suitable.

Chapter 6

The timetabling problem

The model can show the usefulness of minor changes to the arrival and departure times. In fact, shifting the arrival time of a single freight train at Utrecht Centraal by one minute allowed the optimal objective value to decrease by about 50%.

Kroon et al. (2008a)

The train timetabling problem (TTP) is the problem of creating and improving a railway timetable for each train in the network. In the TTP, it is assumed that the line planning with its frequencies per line is given. In this dissertation, the idea is to improve the robustness of a given timetable. As in the previous chapter, the spreading objective function is used to guide the improvement process. One option to tackle the TTP is to extend the Kaufman-TRP model with a time index for each train. However, similar to the observations in Caimi et al. (2005), inserting time options makes the model too arduous to be practical. Also Kroon et al. (2007b) state that solving both problems simultaneously results in models that are too large to be solved with the available technology. Moreover, since the timetabling module is used as subroutine within the platforming module of chapter 7, calling the internal timetabling module should not consume much time. Therefore, the TTP is solved by a heuristic procedure and fixed routes are assumed for each train. As a consequence, the optimal timetable (or the optimal spreading value) as well as the optimality gap are not known.

Since the timetabling module follows upon the routing module, a routing solution is available. Each of these routes connects each train's inbound and outbound lines with its fixed platforms. Because the case studies that are considered in

This chapter is mostly based on Dewilde et al. (2013).

this dissertation are equipped with periodic timetables, only one or two periods (hours) are considered in the algorithm. Upon improving the schedule, trains are not allowed to be shifted to another period. That is why a time window is implied for each train. Where needed, these time windows are used to restrict the deviations from the original schedule or to keep two trains close together, for example, in case of train split or merging actions.

In the next section, the algorithm that is constructed for the TTP is explained. First the outline of the heuristical procedure is given and then the considered timetable changes are introduced. After discussing some features of the timetabling module, the algorithm is tested on the case study of the NSC and the results from this test are presented in Section 6.3.

6.1 The timetabling module

A heuristic is developed for the TTP. During test runs based on a variable neighborhood search, a large threat to cycling and the need for a better guiding of the heuristic became clear. In order to solve these problems, a *tabu search* framework is used. Since it is one of its main properties, tabu search is well suited to avoid cycling. Moreover, several authors, among which Budai-Balke (2009), Corman et al. (2010), Pacciarelli et al. (2001), and Törnquist et al. (2005) used tabu search for timetabling-related problems. In Törnquist et al. (2005), it is shown that tabu search outperforms simulated annealing in some cases. Therefore, the algorithm is embedded in a *tabu search* framework. More details about tabu search can be found in, among others, Gendreau (2003) and Glover et al. (1998).

The tabu search heuristic uses three neighborhood operators or moves: *shift*, *combined shift*, and *order swap*. Without going into details here, each move consists of one or more trains of which the arrival and departure times are changed (shifted) to increase the minimum time span of (at least) one pair of trains[13]. The neighborhoods are ordered from small to large and, similar to the principles of variable neighborhood search, if improvement is found in one neighborhood, the best solution within this neighborhood is selected and the other neighborhoods are not considered.

In the next section, the details of the tabu search algorithm are given and afterwards each of the neighborhood operators is explained.

[13]In the timetabling module, each train has a unique route. Therefore, we talk about trains instead of trainroutes and omit the route-index in our notation.

Algorithm 6.1 Tabu search algorithm of the timetabling module

input: input timetable and routing solution
while stop criteria 1 and 2 are not met **do**
  for $\vartheta = 0..(\varepsilon_{\text{step}})..\vartheta^{\max}$ **do**
    (i) construct the RCL using (6.1)
    (ii) improvement $\leftarrow$ shift
      if ! improvement **then** improvement $\leftarrow$ combined shift
      if ! improvement **then** improvement $\leftarrow$ order swap
    (iii) **if** improvement **then** $\vartheta \leftarrow 0$ and update tabu list
  (iv) make mildest ascent step and update tabu list

6.1.1 Tabu Search

In order to guide the improvement process, the heuristic consists of the four steps of Algorithm 6.1.

(i) The first step is about determining a set of candidate train pairs for which it is expected that an increased minimum time span gives the largest reduction in spreading cost. This set is called the *restricted candidate list (RCL)*. Each time a neighborhood operator is considered, only the elements of the RCL are the candidates for a move. This way, one avoids selecting two trains with a minimum time span larger than $B^{\max}$ since this does not influence the objective function. The RCL is constructed to contain only those pairs of trains of which the minimum time span $B_{t,t'}$ lies within the interval that is bounded from below by the overall minimum time span plus ϑ and has $\varepsilon_{\text{step}}$ as width. Where ϑ is a counter of the algorithm, see the for loop in Algorithm 6.1, $\varepsilon_{\text{step}}$ is a parameter.

$$\text{RCL} = \left\{(t,t') \in T \times T : B_{t,t'} \in \left[\min_{\hat{t},\check{t} \in T} B_{\hat{t},\check{t}} + \vartheta, \min_{\hat{t},\check{t} \in T} B_{\hat{t},\check{t}} + \vartheta + \varepsilon_{\text{step}}\right)\right\}. \quad (6.1)$$

In the following, the upper bound of this interval will be denoted with RCL^{UB}. Thus

$$\text{RCL}^{\text{UB}} = \min_{\hat{t},\check{t} \in T} B_{\hat{t},\check{t}} + \vartheta + \varepsilon_{\text{step}}.$$

(ii) When the RCL is constructed, the search for improvement starts. First, the shift neighborhood is explored. Each time the shift neighborhood, or any of the other neighborhoods, is considered, all elements of the RCL are evaluated and the best one is returned. This principle is called steepest descent. If *improvement* is found, the algorithm proceeds to step (iii) with this new solution. Otherwise, the combined shift neighborhood is investigated. Here, the same principle is repeated and, if no solution is accepted, an order swap is considered. When this did not result in a better timetable, the search returns to step (i) with $\vartheta \leftarrow \vartheta + \varepsilon_{\text{step}}$ as long as $\vartheta \leq \vartheta^{\max}$. According to (6.1), the increase of ϑ implies an update of the RCL which then contains the pairs of trains with a slightly larger minimum time span. If ϑ becomes too large, $\vartheta > \vartheta^{\max}$, no better solution is found for any considered train pair. In this case, the for loop ends and the algorithm proceeds to step (iv) with the best, non-tabu solution that is found since the last move was made. Allowing the best non-improving neighboring solution is called the principle of mildest ascent.

(iii) and (iv) The algorithm reaches step (iii) when an improving move is made. If, within the entire for loop of ϑ, no improving move is found, this for loop ends and the algorithm reaches step (iv). In this step, a mildest ascent move is made. In both step (iii) and step (iv), the timetable is replaced by a new timetable. To prevent the algorithm from returning to the previous solution, *tabu lists* are used. In such a list, the reverse of each move is stored for a number of iterations (*tabu tenure*). During these iterations, making this reverse change is tabu. In the timetabling module, two independent tabu lists are used. The first contains the reverse shifts (independent of the size of the shift). The second keeps track of changes in the order of trains. To illustrate this, denote the size of the time shift for train t with δ_t and consider a move that alters the timing of trains t and t' with $\delta_t > 0$ and $\delta_{t'} < 0$ such that this incurs a swap in the order of these trains. Then shifting t (t') with a negative (positive) δ or restoring the original order of these trains is tabu. Except if it improves the globally best solution, tabu moves are forbidden. This exception is called the aspiration criterion. The size of the deterministic, but list-dependent tabu tenures that are used is discussed in Section 6.3. After the tabu list update, the algorithm returns to step (i) with $\vartheta = 0$ unless the maximum number of moves (stop criterion 1) or the maximum number of consecutive, non-globally improving moves (stop criterion 2) is reached in which case the algorithm terminates.

A commonly used feature of tabu search is a diversification loop. In our tabu search algorithm, there is no explicit diversification phase but the platforming module of the next chapter and the routing module are used to diversify the search.

6.1.2 The shift operator

In order to explore the shift neighborhood, each pair of trains in the RCL is considered one by one. In this and the following sections, the RCL element is called the *active* pair and is indicated with (t, t'). Using this notation, it is assumed that t precedes t' in the critical block section of these two trains. Next to the usage of δ_t for the time shift of train t and the notation $t + \delta_t$ to indicate the shifted train, (t, δ_t) is used to refer to the shift of t with δ_t.

A shift move, the most basic timetable change of the algorithm, consists of the shift of one or more trains in time with $\delta_t = -\delta_{\min}..(1)..-1$[14] or $\delta_{t'} = 1..(1)..\delta^{\max}$. The values of the model parameters $\delta_{\min}$ and $\delta^{\max}$ (and the other parameter values) are discussed in Section 6.3. Since t precedes t', the shifts $(t, \delta_t < 0)$ and $(t', \delta_{t'} > 0)$ increase $B_{t,t'}$ by advancing train t or postponing t'. Only in case the trains have opposing orientations and the shift changes the critical block section, the size of the minimum time span can decrease. The two possible situations are illustrated in Figures 6.1 and 6.2. These figures represent the time-space diagram of the two trains that pass through the illustrated network[15]. For the block sections where the infrastructure is shared by both trains, the blocking times are drawn. In both figures, train t gets shifted with a negative δ_t. Figure 6.1 illustrates the standard case with increasing minimum time spans. Figure 6.2 shows the special case in which the critical block section changes and causes a reduction in the size of the minimum time span between t and t'. If the (absolute value of) the shift size would be smaller or if no infrastructure would be shared in block sections bs_3 and bs_4, there would be no shift of the critical block section and then the shift of t with δ_t would be beneficial.

The shift of a train can decrease the minimum time span between this train and some other trains. To avoid conflicts, some of these trains need to be shifted as well. The procedure to determine the set of trains that will be shifted, say ST, is given in Algorithm 6.2. Let (t, δ_t) be the shift move under consideration.

Algorithm 6.2 Procedure to compute the set of trains that will be shifted (ST)

input: (t, δ_t) with (t, t') the active RCL pair.
$ST \leftarrow \{t\}$
for all $t \in ST$ **do**
  for all $\bar{t} \in T$ with $B_{\bar{t},t} > B_{\bar{t},t+\delta_t}$ **do**
    (i) **if** $B_{\bar{t},t+\delta_t} \leq 0$ or the order of t and $\bar{t}$ at their (initial) cricital block section reverses **then** $ST \leftarrow ST \cup \{\bar{t}\}$
    (ii) **if** $0 < B_{\bar{t},t+\delta_t} < B_{t,t'}$ **then**
      $v \leftarrow rand\{0..(1)..B_{t,t'}\}$
      if $v = 0$ **then** $ST \leftarrow ST \cup \{\bar{t}\}$
    (iii) **if** $B_{t,t'} \leq B_{\bar{t},t+\delta_t}$ **then** do not add $\bar{t}$ to ST

[14] In this dissertation, the notation $a..(b)..c$ is used to indicate the elements $a, a+b, a+2b, \ldots, a+ib$, with i the largest integer for which $a + ib \leq c$.

[15] For a good introduction about blocking time theory and the time-space diagram representation, we refer to Pachl (2008).

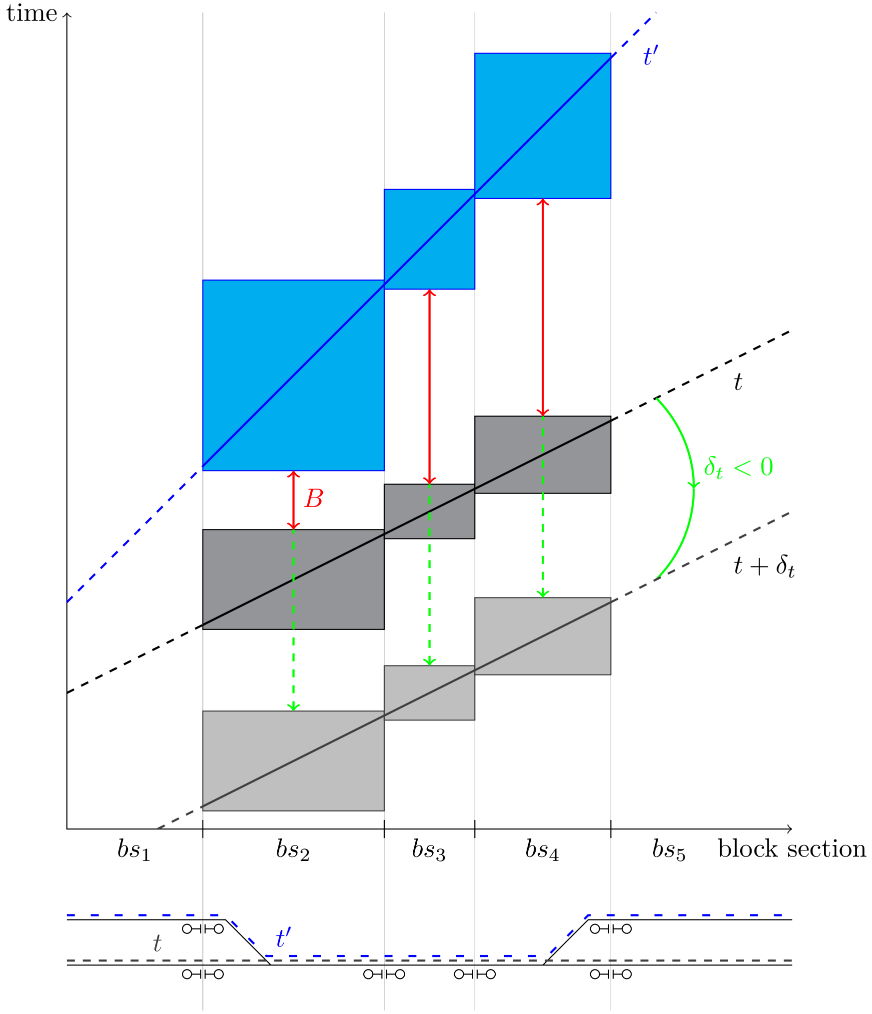


Figure 6.1: Illustration of a shift move that increases the minimum time span between two trains. In this time-space diagram, the trajectories of trains t and t' that pass through the illustrated network, are indicated with the full (dashed) line for the block sections where the infrastructure is (not) shared by both trains. The impact of the shift of t with $\delta_t < 0$ results in larger time spans between the gray $t + \delta_t$ and t' at all block sections. The increase is indicated with the dashed arrows.

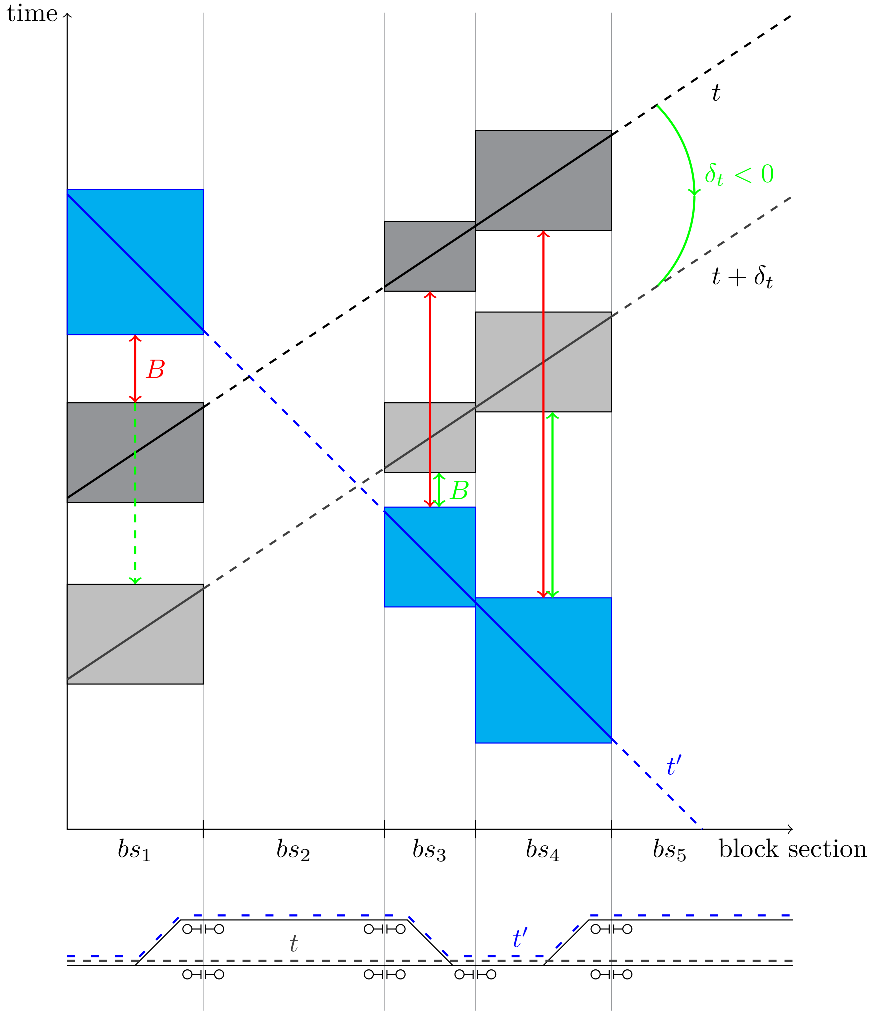


Figure 6.2: Illustration of a shift move that decreases the minimum time span between two trains. In this time-space diagram, two trains with opposing orientations share some infrastructure. Due to the shift of train t with $\delta_t < 0$, block section bs_3 becomes critical instead of bs_1. Since the order of the two trains at the new critical block section differs, the shift move decreases the minimum time span.

Initially, this set contains only train t. Then for each element of ST, for simplicity call it t, and each train $\bar{t}$ of which the minimum time span with this train decreases due to the shift, $B_{\bar{t},t} > B_{\bar{t},t+\delta_t}$, the following three situations are distinguished.

(i) If the decreased minimum time span incurs a conflict, then $\bar{t}$ is added to ST and gets shifted with δ_t. In Figure 6.3 this situation is illustrated. In this figure, the shift (t, δ_t) is made to increase $B_{t,t'}$. Since the blocking times of $t + \delta_t$ and $\bar{t}$ overlap, a conflict arises. In order to solve this conflict, $\bar{t}$ is shifted along with t and $\delta_{\bar{t}} = \delta_t$. Also in case the order of t and $\bar{t}$ at their (initial) critical block section reverses, $\bar{t}$ is added to ST. Notice that in the example of Figure 6.2, the critical block section changes due to the shift, but the order at the initial critical block section (bs_1) did not change. As a consequence, t' is not added to ST in this example.

If $\bar{t}$ is added to the set of shifted trains, the outer for loop in Algorithm 6.2 ensures that all decreased minimum time spans between $\bar{t}$ and any other train are considered as well.

(ii) The second situation is the one where no conflict arises but $B_{\bar{t},t+\delta_t}$ becomes smaller than the minimum time span of the active pair before the shift, $B_{t,t'}$, and thus also smaller than $\mathrm{RCL}^{\mathrm{UB}}$, the upper bound of the RCL. In this case, the decision of whether $\bar{t}$ is added to the list of shifted trains depends on a random factor. Let v be a random integer that is drawn from the uniform distribution such that $0 \leq v \leq B_{t,t'}$, then $\bar{t}$ is shifted with $\delta_{\bar{t}} = \delta_t$ if v equals 0. Thus, the smaller the initial minimum time span, the larger the chance that $\bar{t}$ shifts along with t. This way, critical time spans are avoided and the chance that the entire move is improving increases. During the design of the algorithm, the impact of both extremes, always shift $\bar{t}$ or never shift it, are considered. Since none of the two outperformed the other, we opted for this intermediate solution.

(iii) In the third situation, the decreased minimum time span remains larger than $B_{t,t'}$. Since the impact on the spreading cost of this decreased minimum time span is expected to be smaller than the gain of increasing $B_{t,t'}$, $\bar{t}$ is not shifted in this case.

For each pair of trains in the RCL and for all shift sizes, the set ST of trains that will be shifted is computed. The shift of all these trains is evaluated by computing the spreading cost of the new timetable. If the shift of one of the selected trains is tabu, the shift move is labeled tabu and may not be performed, except if it improves the currently best solution. Note that only one of the two active trains (t or t') is allowed to be shifted and changes in the order of two trains, except for situations like in Figure 6.2, are forbidden.

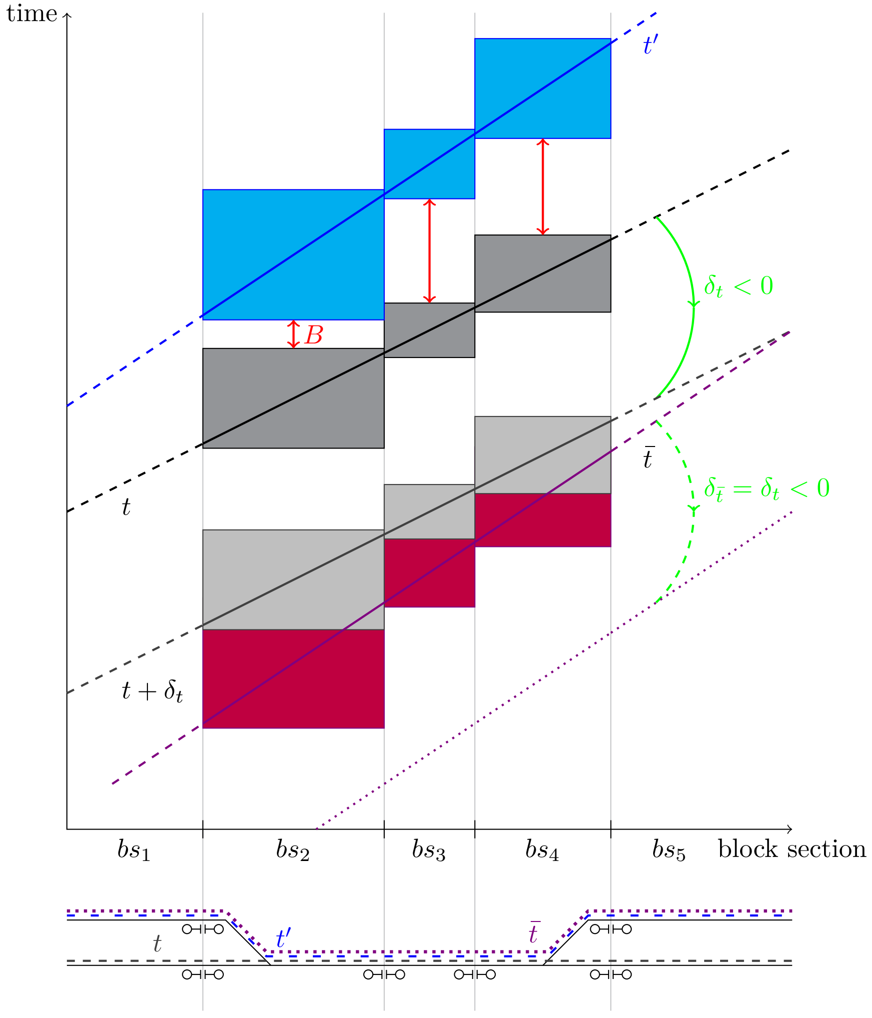


Figure 6.3: Example of a shift triggering other shifts. The RCL candidate is (t, t') and (t, δ_t) is the shift that is tested. Since $B_{\bar{t},t} < |\delta_t|$, the shifted train $t + \delta_t$ conflicts with $\bar{t}$. Therefore, also $\bar{t}$ needs to be shifted (dashed arrow).

6.1.3 Combined Shift

Where the shift operator only allows a shift in one direction with size δ, the combined shift operator considers the combination of multiple shifts to explore a larger neighborhood. It starts with a simple shift, then the RCL is updated according to a specific rule, and a second shift move is performed. This process is repeated until $CS^{\max}$ shift moves are made. After each move, the spreading cost of the new timetable is evaluated, and at the end, the best of all candidates is selected. This is not necessarily the one in which $CS^{\max}$ individual shifts are made.

A typical example of a combined shift move is illustrated in Figures 6.4 and 6.5. In Figure 6.4, there are three trains that run through the depicted network. Assume that the time window constraint of train $\bar{t}$ does not allow that $\bar{t}$ is shifted with $\delta_{\bar{t}} < 0$. The two possible moves to improve this situation are the shift (t, δ_t) and $(t', \delta_{t'})$ with δ_t and $\delta_{t'} > 0$. Notice that the latter shift pushes t to be shifted too with $\delta_t = \delta_{t'}$. These two options are depicted in Figure 6.4. In case none of these individual shifts (for any value of δ_t or $\delta_{t'}$) reduces the total spreading costs, for example, due to other trains in the network, then the combined shift that is indicated in Figure 6.5 can be useful. For example, by selecting $\delta_{t'} = \delta_t/2$, two shifts of different sizes are made and the spreading cost of the three trains will be smaller than when any single shift move is performed. Therefore, the combination of (t, δ_t) and $(t', \delta_{t'})$ can be enough to get improvement.

The kernel of the combined shift operator is the RCL update that follows upon a shift. The idea is to keep the set of candidate moves related to the previous shifts. This way the combined shift of two trains which are far apart in time is avoided. The three step procedure to update the RCL is summarized in Algorithm 6.3. Similar to the definition of ST as the set of shifted trains, let ST^k be the set of trains that are shifted in the k^{th} shift. As is explained in the previous section, this set contains at least one train but often, more trains are shifted. Let $B^0_{t,t'} = B_{t,t'}$. For the minimum time span between trains t and t', $B^{k-1}_{t+\delta^k_t,t'} = B^k_{t,t'}$ is used to indicate its size after the k^{th} shift with shift size δ^k is made.

The RCL update procedure goes as follows.

(i) After the first shift of a combined shift move, the RCL is emptied. This is done to avoid the shift of two trains which are far apart in time. Although both shifts increase a minimum time span that is smaller than RCL^{UB}, if they are separated by, for example, half an hour, it makes no sense to apply these two shifts simultaneously. For later shifts, $k > 1$, the RCL contains only train pairs

Algorithm 6.3 Procedure to update the restricted candidate list (RCL) during combined shift

input: k, RCL, ST^k, B^k, RCL^{UB}
(i) **if** $k = 1$ **then** RCL $\leftarrow \emptyset$
  else $\text{RCL} \leftarrow \text{RCL} \setminus \left\{(t,t') \in \text{RCL} : B^k_{t,t'} \geq \text{RCL}^{\text{UB}}\right\}$
(ii) **for all** $(t,t') \in T \times T$ **do**
  if t or $t' \in ST^k$ and $B^k_{t,t'} < \text{RCL}^{\text{UB}}$ **then** $\text{RCL} \leftarrow \text{RCL} \cup \{(t,t')\}$
(iii) **for all** $(t,t') \in ST^k \times T$ **do**
  if $B^{k-1}_{t,t'} < \text{RCL}^{\text{UB}}$ and $B^{k-1}_{t+\delta^k_t,t'} = B^k_{t,t'} \geq \text{RCL}^{\text{UB}}$ **then**
   for all $t'' \in T$ **do**
    if $B^k_{t',t''} < \text{RCL}^{\text{UB}}$ and $B^k_{t',t''} < B^k_{t'+\delta^k_t,t''}$
    then $\text{RCL} \leftarrow \text{RCL} \cup \{(t',t'')\}$

that are related to one of the previous shifts. Thus, there is no need to clear this list then. Only the pairs of trains of which the new minimum time span is at least RCL^{UB} are removed from the RCL. This is step (i) in Algorithm 6.3.

(ii) In the second step, the RCL is extended with all pairs of trains (t,t') for which t or t' has been shifted in the k^{th} shift (t or $t' \in ST^k$) and for which the minimum time span after this shift is smaller than RCL^{UB} ($B^k_{t,t'} < \text{RCL}^{\text{UB}}$). There are three situations in which the latter condition is satisfied.

(a) If the initial minimum time span, $B^0_{t,t'}$, is smaller than the lower bound in (6.1) and both trains are shifted simultaneously. Notice that this pair of trains was part of the original RCL but is removed in step (i) of the RCL update that followed the first shift.

(b) Due to the random influence within the shift operator. Let $(t,\bar{t})$ be the pair of trains that initialized the k^{th} shift and assume that t is shifted with δ^k_t. If this shift makes the minimum time span between t and t' smaller than the time span that is supposed to be increased by this shift,

$$0 < B^{k-1}_{t+\delta^k_t,t'} < B^{k-1}_{t,\bar{t}}, \tag{6.2}$$

then the random parameter υ determines whether t' is shifted with δ^k_t. This corresponds to situation (ii) of the ST construction procedure of a shift move. If t' is not added to ST^k, then the inequalities in (6.2) remain valid. Since the previous shift is made to increase $B^{k-1}_{t,\bar{t}}$, this minimum time span is smaller than RCL^{UB}. As a consequence, the new minimum time span between t and t' is also smaller than RCL^{UB}.

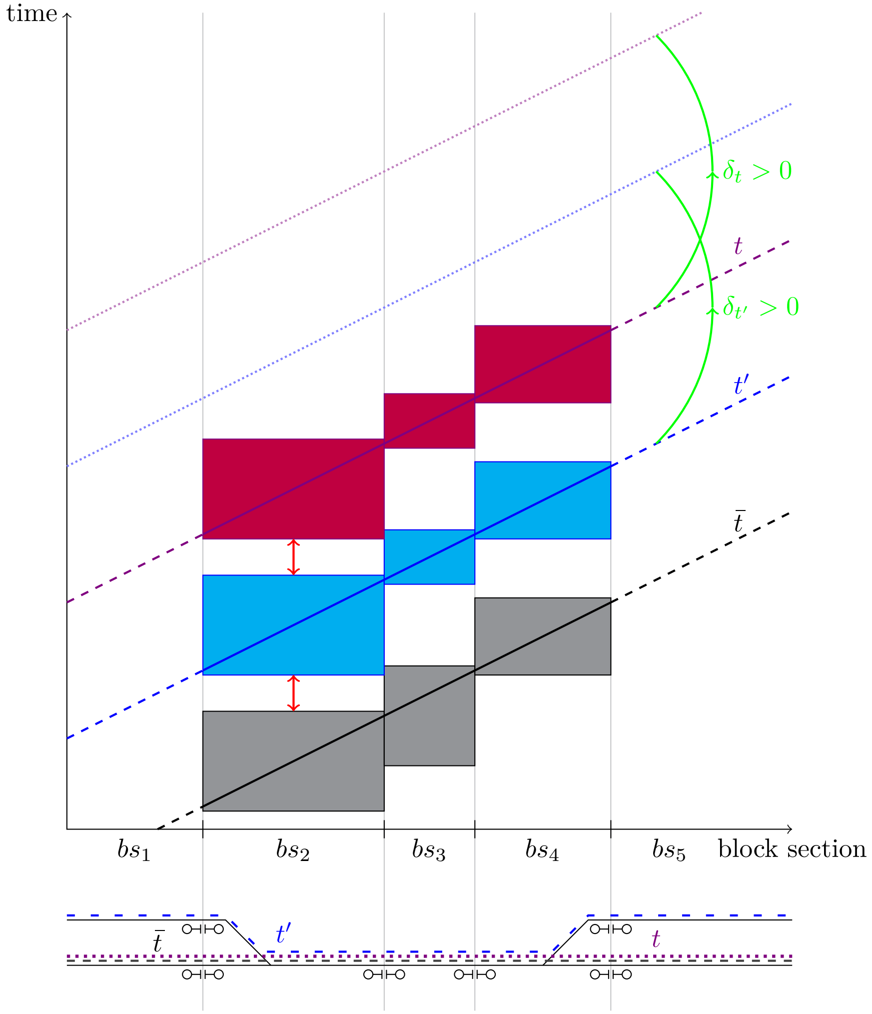


Figure 6.4: Example of a combined shift. Overview of the initial situation with three trains and the allowed shift possibilities.

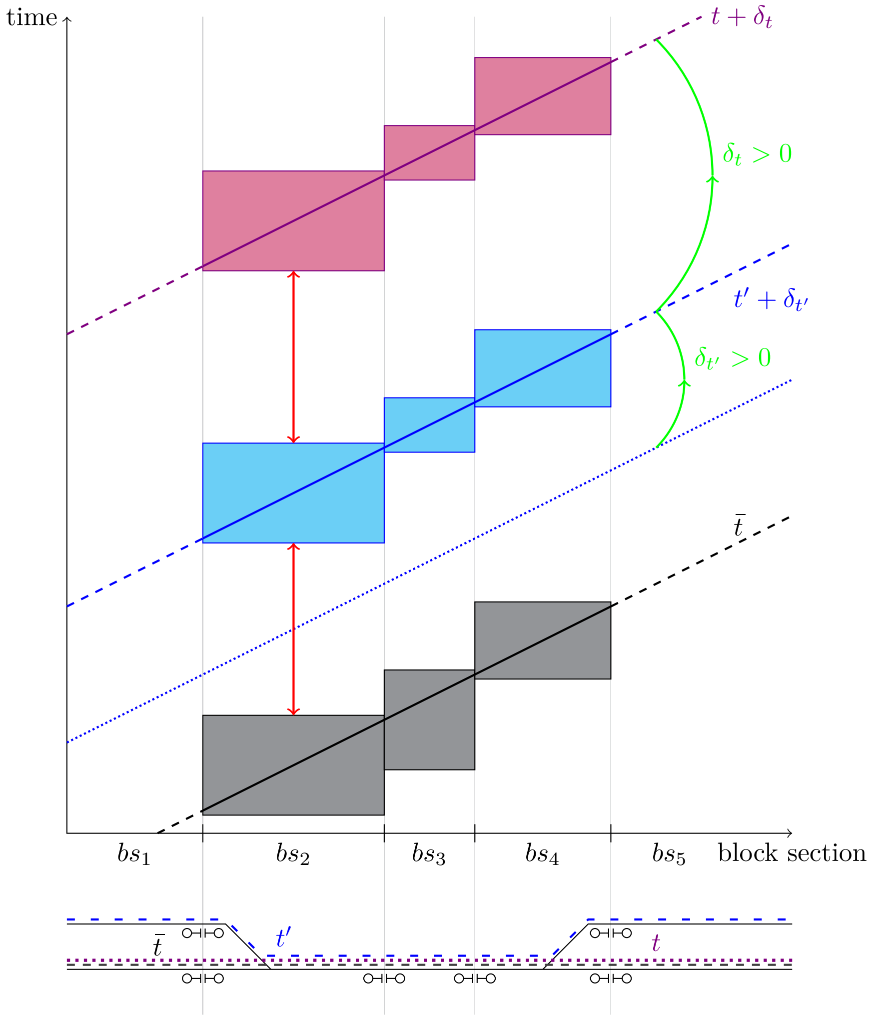


Figure 6.5: Example of a combined shift. The result of the combined shift move. There are two ways to obtain this result. The first is when t is shifted first with $\delta_t > 0$ and then $(t', \delta_{t'})$ follows. The second way is by shifting t' and t with $\delta_{t'}$ and then applying an extra shift for t with size $\delta_t - \delta_{t'}$.

(c) When the shift of t decreases the minimum time span between t and t' such that this becomes smaller than $\mathrm{RCL}^{\mathrm{UB}}$ and matches situation (iii) of Algorithm 6.2 ($B^{k-1}_{t,\bar{t}} \leq B^{k-1}_{t+\delta^k_t,t'}$), then t' is not shifted together with t (during the k^{th} shift) and thus $B^k_{t,t'} < \mathrm{RCL}^{\mathrm{UB}}$.

(iii) In the third step of the RCL update, pairs of trains are added to the RCL based on the enabled possibility for improvement by prior shifts. The property of a move to enable further improvements is what we call the *potential* of a move. The k^{th} shift can have enabled the possibility to increase the minimum time span of a non-shifted train pair. For any $t \in ST^k$ and $t' \in T$ for which the shift of t with δ^k_t made their minimum time span larger than $\mathrm{RCL}^{\mathrm{UB}}$, $B^{k-1}_{t,t'} < \mathrm{RCL}^{\mathrm{UB}}$ and $B^{k-1}_{t+\delta^k_t,t'} = B^k_{t,t'} \geq \mathrm{RCL}^{\mathrm{UB}}$, there could be a train $\bar{t}$ for which $B^k_{t',\bar{t}} < \mathrm{RCL}^{\mathrm{UB}}$ and the shift of t' in the same direction as t is shifted in the previous iteration increases $B^k_{t',\bar{t}}$. Although this shift goes at the cost of a smaller minimum time span between t and t', the total spreading cost may reduce since for $|\delta^{k+1}_{t'}| < |\delta^k_t|$, the following inequalities

$$0 < B^k_{t',\bar{t}} \leq B^k_{t'+\delta^{k+1}_{t'},\bar{t}} \leq B^k_{t,t'}$$

and

$$0 < B^k_{t',\bar{t}} \leq B^k_{t,t'+\delta^{k+1}_{t'}} \leq B^k_{t,t'}$$

imply that

$$C^k_{t',\bar{t}} + C^k_{t,t'} \geq C^k_{t'+\delta^{k+1}_{t'},\bar{t}} + C^k_{t,t'+\delta^{k+1}_{t'}}.$$

Therefore, $(t', \bar{t})$ is added to the RCL.

The combined shift of Figures 6.4 and 6.5 illustrates the usefulness of this step. Thanks to the shift $(t, \delta_t > 0)$ which makes $B^{k-1}_{t+\delta^k_t,t'}$ larger than $\mathrm{RCL}^{\mathrm{UB}}$, the minimum time span between t' and $\bar{t}$ can be increased by shifting t' with $\delta_{t'} > 0$.

The updated RCL is used to determine shift $k+1$. After each shift, the combined shift is evaluated based on the total spreading cost of the new timetable. During each of the shifts, the tabu status is ignored and only determined for the combined move. Doing so, the combined shift of train t with $\delta^k_t > 0$ and $\delta^{k'}_t < 0$ for $k \neq k'$ is non-tabu as long as the shift $(t, \delta^k_t + \delta^{k'}_t)$ is non-tabu. When $k = CS^{\max}$, the best solution of the $CS^{\max}$ candidates is selected. If this solution improves the globally best solution or improves the current solution without being tabu, it is accepted. Otherwise, the move is rejected.

6.1.4 Order Swap

The shift and combined shift operators do not allow a change in the order of trains at the initial critical block section. If the shift of one train would overtake another train at this block section, the other train is also shifted. This is in contrast with the order swap operator which aims at reversing the order of two trains. After such a swap, exploiting the potential can be very beneficial for the optimization. Moreover, using the freed time slots for other trains often is necessary to obtain a better spreading cost.

For example, in Figure 6.6, the time-space diagram is drawn for the three trains that run over the depicted network. For the block sections with shared resources, the time spans between the trains are indicated with the two-sided arrows and the minimum time span is named. Notice that $B_{t,\bar{t}} = \infty$ since the routes of these trains are parallel. When neither a shift nor a combined shift can improve the spreading cost of this situation, the order swap of trains t' and $\bar{t}$ is investigated. To swap the order of these trains at their critical block section bs_2, t' is shifted with $\delta_{t'} > 0$ and $\bar{t}$ with $\delta_{\bar{t}} < 0$. The result of the order swap is shown in Figure 6.7. Clearly, this swap has improved the spreading cost since the two minimum time spans are increased. Moreover, there is a large potential for further improvement. Since trains t and $\bar{t}$ do not hinder each other when they run through the network, the minimum time span between $\bar{t}$ and t' can be increased even more by applying a second shift for $\bar{t}$ like indicated in Figure 6.7 with $\delta^2_{\bar{t}} < 0$.

The methodology behind the order swap operator is summarized in Algorithm 6.4. An order swap consists of five steps.

(i) The first step is the selection of the swap pair. Let (t, t') be the RCL-candidate with t preceding t' at their critical block section. Denote with $\underline{t}$ ($\bar{t}$) the train closest to t (t') such that $\underline{t}$ precedes t and t' precedes $\bar{t}$. One by one, the order swap of the pairs $(\underline{t}, t)$, (t, t'), and $(t', \bar{t})$ is considered. All three

Algorithm 6.4 Details of the order swap

input: RCL
for all $(t, t') \in$ RCL **do**
  (i) determine $\underline{t}$ and $\bar{t}$
  for $(\underline{t}, t), (t, t')$, and $(t', \bar{t})$ **do**
    (ii) determine δ^{tot}
    (iii) **for** $\delta_t = 0..(\delta_{\text{step}})..\delta^{\text{tot}}$ and $\delta_{t'} = \delta_t - \delta^{\text{tot}}$ **do**
      (iv) shift t and t' or the whole extended set
      (v) perform the internal timetabling module

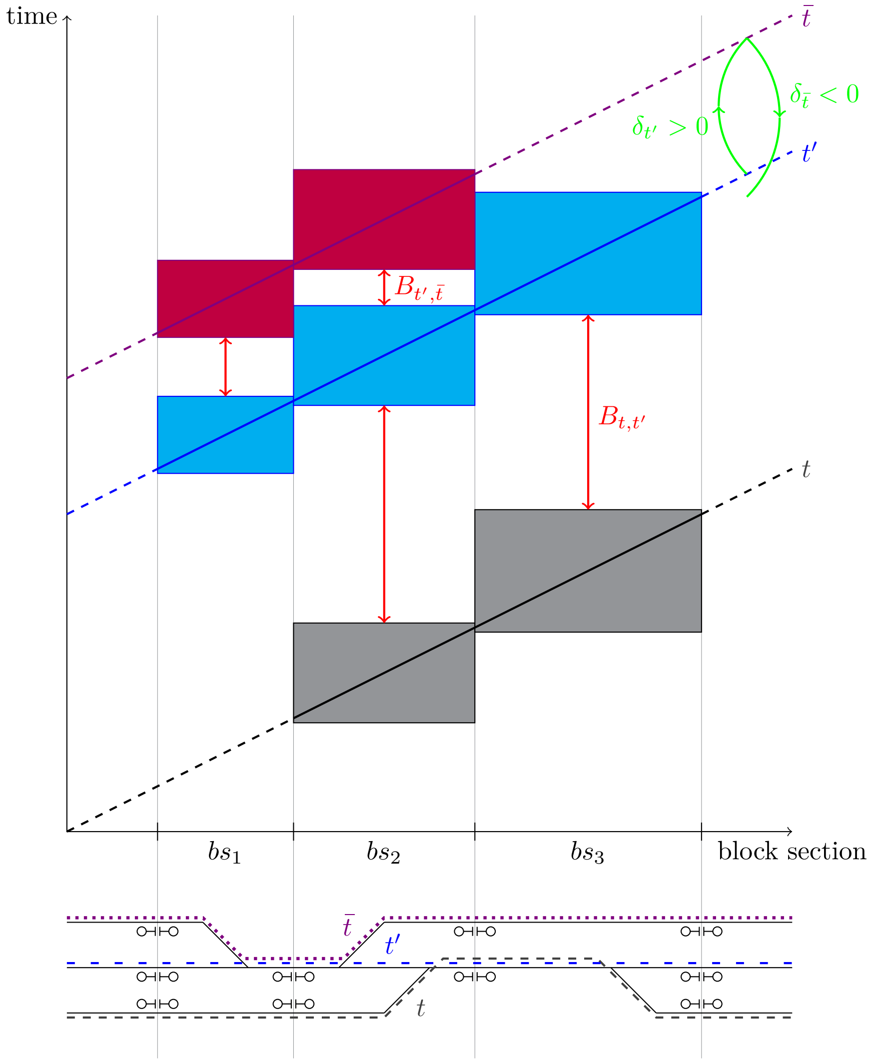


Figure 6.6: Illustration of an order swap. In the time-space diagram, the course of the three trains that run through the network is plotted. Comparing the routes of each train, one sees that trains t and t' share block sections bs_2 and bs_3, while t' and $\bar{t}$ use equal resources in bs_1 and bs_2. The routes of t and $\bar{t}$ are parallel. To swap the order of t' and $\bar{t}$, $\delta_{t'} > 0$ and $\delta_{\bar{t}} < 0$ are used. The resulting time-space diagram is depicted in Figure 6.7.

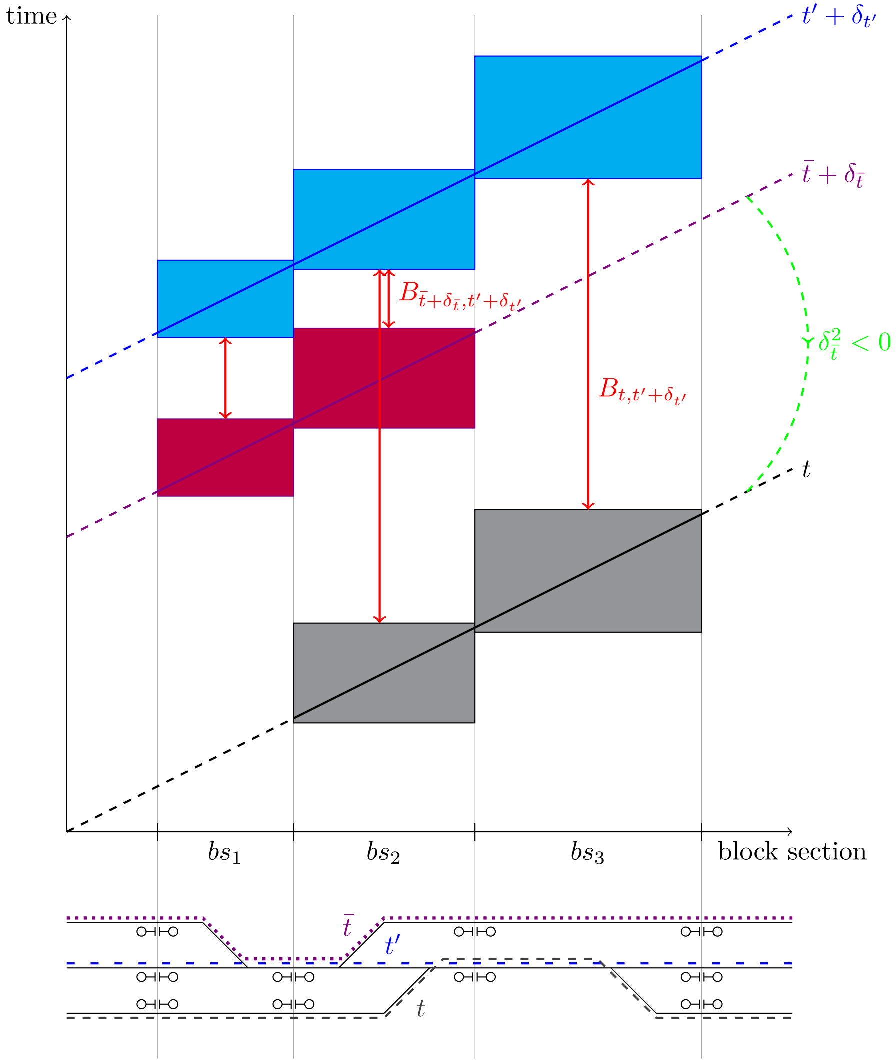


Figure 6.7: Illustration of the potential of an order swap. After the order swap of Figure 6.6, a potential for further improvement has arisen. Since t and $\bar{t}$ do not share any resources, $\bar{t}$ can be shifted with $\delta^2_{\bar{t}} < 0$ (dashed arrow) such that it runs simultaneously with t through the network. This increases the minimum time span between $\bar{t}$ and t' considerably.

go through steps (ii)-(v). This is done because of the observation that sometimes $B_{t,t'}$ cannot be increased without involving other trains. For example, if $B_{t,t'}$ would be smaller than $B_{t',\bar{t}}$ in Figure 6.6 and $(t', \delta_{t'} < 0)$ is not allowed, then the order swap of t' and $\bar{t}$ can help to improve this situation. For the ease of notation, (t, t') is used as swapping pair and it is assumed that t precedes t'.

(ii) In the second step, the size of the total shift, δ^{tot}, is determined. This is the smallest value for which $t + \delta^{\text{tot}}$ overtakes t' and $B_{t+\delta^{\text{tot}},\,t'} \geq B_{t,t'}$.

(iii) After that, the selected values for δ_t and $\delta_{t'}$ are considered one by one in step (iii). This is a for loop that starts with $\delta_t = 0$ and $\delta_{t'} = -\delta^{\text{tot}}$ and adapts the shift sizes in each iteration with δ_{step} such that $\delta_t + |\delta_{t'}| = \delta^{\text{tot}}$ at all times.

(iv) For each combination of δ_t and $\delta_{t'}$, step (iv) is performed twice. The first time, only t and t' are shifted before the algorithm proceeds to step (v). The second time, the set of shifted trains is extended. The selection criteria for a train to be shifted along with t or t' are analogous to the ones of the shift operator, but the size of the shift can be the smallest (in absolute values) that solves the conflict. For half of the computational tests, only shifting t and t' outperformed shifting the extended set. For the other half, however, this was vice versa. Since no clear rule about the difference in performance is deduced, both versions are kept in the algorithm.

(v) When each selected train is shifted, there can still be a conflict somewhere. In the first iteration of step (iv) this is expected, but also in the second iteration this can occur. Although trains should be added to avoid conflicts, this is not always possible. For example, when the shift causes a train to leave its predetermined time window or when different conflicts require a train to be shifted in opposite directions. In case the new timetable is feasible or can be made feasible, several very narrow headway buffers can arise and thus an improved spreading cost cannot be expected immediately. Therefore, one has to consider the possibilities for further improvement that are enabled by the order swap. To restore feasibility and to explore the potential of the order swap, step (v) is a recursive call to the timetabling module which we name the *internal timetabling module*. Where the RCL is normally built using (6.1), it is the set of conflicting trains that forms the RCL in case of an infeasible solution. Since the algorithm tries to increase the minimum time span of each RCL pair, the internal timetabling module is appropriate to solve conflicts or to improve the solution. Note that the order swap neighborhood itself is excluded from the internal timetabling module. The stopping criteria are more restrictive than in the outer timetabling module. If no conflict is solved after a number of iterations or if a maximum number of moves are made, the internal timetabling module stops and returns a solution to the order swap.

Each solution that is returned from the internal timetabling module is evaluated on three levels: feasibility, its tabu status, and the spreading cost. The tabu status of the entire order swap move is determined using the two tabu lists. The move is labeled tabu if the swapped order is tabu based on the second tabu list or if the total shift of one of the trains is tabu. In the end, the order swap operator returns the best, feasible, non-tabu or globally improving solution that is found after the internal timetabling module is applied for all elements of the RCL.

Using the recursive call to the (internal) timetabling module has a large impact on the performance of the algorithm since it allows to work with infeasible solutions and to explore the potential of a move. In the example of Figures 6.6 and 6.7, the swapped trains merit from the potential. Sometimes, also other trains benefit from the swap. For example, in Figure 6.8, the swap of t and t' with $\delta_{t'} = -\delta^{\text{tot}}$ is performed. Since $B_{t'+\delta_{t'},t}$ should be larger than $B_{t,t'}$, a large shift for t' is needed. As a consequence, the order of t' and $\hat{t}$ gets swapped too such that t' does not block the shift of $\hat{t}$ with $\delta_{\hat{t}} > 0$ anymore. The improvement that can be gained by this shift, is part of the potential of the order swap of t and t'.

6.2 Framework and discussion

An overview of the framework is given in Algorithm 6.5. The timetabling module to tackle the TTP always follows the routing module. Based on a tabu search heuristic, the timetable is improved step by step. In the end, the timetable with the smallest spreading cost forms the output. For this timetable, the TRP is solved again and then the timetabling module is repeated. In the results of the next section, the algorithm ends in case no improvement is achieved at the end of the tabu search heuristic. Thus for now, $iter^{\max}$ is set to 0 and the if-clause at the end of Algorithm 6.5 is ignored. In the full version of the algorithm, however, the platforming module of the next chapter is called instead.

Algorithm 6.5 Framework of the developed algorithm: the timetabling module

input: infrastructure data and reference timetable
while number of consecutive non-improving iterations $\leq iter^{\max}$ **do**
  solve the train routing problem (TRP)
  apply tabu search (timetabling module)
  if timetabling did not yield improvements
    then start the platforming module

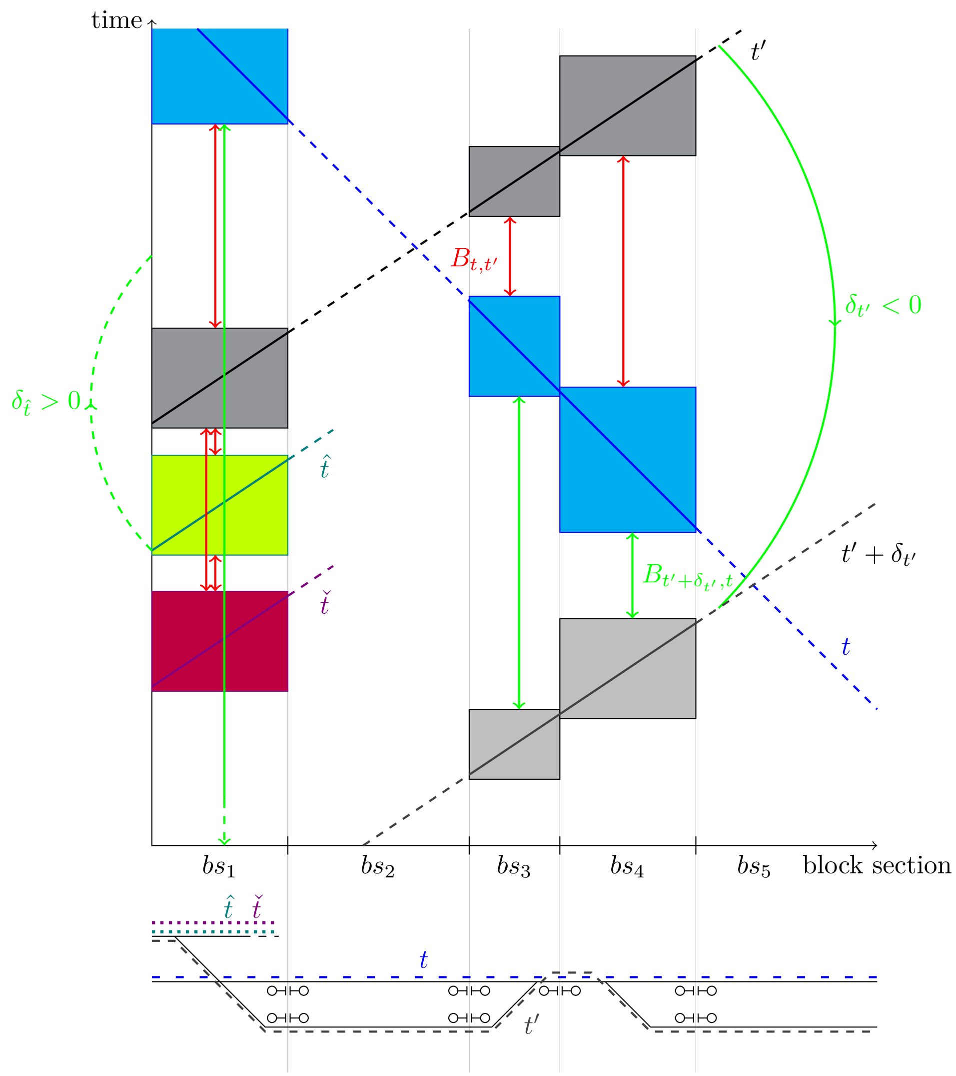


Figure 6.8: Example of an order swap where other trains benefit from the created potential. After the swap of t and t', the minimum time span between $\check{t}$ and $\hat{t}$ can be increased since t' does not block the shift $(\hat{t}, \delta_{\hat{t}})$ anymore.

At the start of the tabu search algorithm, the reference timetable or the one from the previous iteration as well as a feasible routing solution is available. Since moves are only accepted if they are conflict-free (after the internal timetabling module), the routing solution remains feasible throughout the entire tabu search run. This ensures that a succeeding call to the routing module always finds a solution. Moreover, the current routing solution can be used for a warm start. That is, the best spreading cost serves as upper bound for the new objective function of the optimal routing solution.

Most studies about the timetabling problem consider a macroscopic network or focus on one or more lines. Microscopic TTP-like studies on station areas are not that numerous. For example, Caimi et al. (2005) and Zwaneveld et al. (1996) allow limited timetabling options when solving the TRP for busy and complex networks. Scheduling and routing for a set of stations on the line is done in, among others, Carey et al. (2007).

Some of the main difficulties one faces when solving the timetabling problem are the ordering decisions. For example, the phase shifts, the variables related to the cyclic order of events, make a PESP model hard to solve. Therefore, similar to what is done in Kroon et al. (2008d), ordering decisions often are predetermined and kept fixed during optimization. Also in our approach, the reference timetable provides the initial ordering. However, during the timetabling module, order swaps are performed to improve the sequence of events. By testing the potential of the new order, well thought ordering decisions are made. This goes at the cost of some computation time since the recursive function to test the potential of a swap requires several calls to the other neighborhood operators. However, at the tactical level planning phase, the importance of computation times are overruled by the quality of the result, in this case, the spreading and/or robustness of the timetable.

Measuring the quality of a timetable by some spreading function has been done by some other authors. The spreading objective function (4.3) is similar to the sum of the shortest headway reciprocals (SSHR) that is introduced by Vromans et al. (2006). Using the sum of the reciprocals, the impact of heterogeneity on the reliability of a railway system can be measured since headway buffers at the beginning and end of a track section will differ due to speed differences. Where the applicability of the SSHR measure in Vromans et al. (2006) is restricted to track sections between stations, we consider the stations and use (4.3) as objective function instead of a reliability measure. For more references concerning the SSHR or other, related analytical measures, the reader is referred to Andersson et al. (2013), Lindfeldt (2013), and Salido et al. (2012), and the references contained therein.

6.3 Results

Based on some preliminary testing, the values of the parameters that are introduced in the algorithm are determined. An overview of all these parameters and their selected values is given in Table 6.1. Although minimum time spans smaller than $B^{\max} = 15$ minutes incur a spreading cost, the upper bound of the RCL with width $\varepsilon_{\text{step}} = 0.5$ minutes, is 5 minutes ($\vartheta^{\max}$). Due to the high interaction rate within the station area, no benefit is found by considering larger time spans. Moreover, the impact of mildest ascent moves for large time spans fades. Next, the maximal timetable changes within the shift neighborhood, $\delta_{\min}$ and $\delta^{\max}$, are restricted to 5 minutes. This means that in the (internal) timetabling module, all shifts of 5 minutes or smaller are considered for each RCL candidate. The value of 5 is selected since it gave the best results during test runs of the algorithm in which all integer values between 1 and $B^{\max}$ were tested. To avoid the rehearsal of too many shifts in the internal timetabling module, the evolution of the shift sizes in the order swap neighborhood goes in steps of 5 minutes (δ_{step}). Since the optimal number of combined shift moves did not rise above 10, this value is used for $CS^{\max}$. The two tabu tenures are set to 5 iterations. This means that, after each (combined) shift move, all reverse shifts remain tabu for the next five moves. Similarly, undoing an order swap is only allowed after five other swaps, except if it improves the globally best solution. In order to obtain these values, the impact of various tabu tenures is compared. Static as well as dynamic tabu tenures of varying sizes are tested and based on the average performance, the static value of 5 is selected for both tabu lists. The last values in Table 6.1, are the stop criteria. The first criterion stops the algorithm if no improvement of the best solution was generated in the last 200 (5) iterations in the regular (internal) timetabling module. If the timetable is infeasible, this also means that each 5 steps, at least one conflict should be solved. The second stop criterion bounds the total number of moves that are made and is only used in the internal timetabling module.

Table 6.1: Parameter values for the timetabling module.

Parameter	Value (min)	Parameter	Value (iter)
$B^{\max}$	15	$CS^{\max}$	10
$\varepsilon_{\text{step}}$	0.5	shift-tabu tenure	5
$\vartheta^{\max}$	5	swap-tabu tenure	5
$\delta_{\min}$	-5	stop criterion 1	200 (5)
$\delta^{\max}$	5	stop criterion 2	∞ (30)
δ_{step}	5		

After three iterations of the combination of routing and timetabling, no improvements are found anymore. In comparison with the reference timetable, 74 trains, on a total of 80, are shifted with 4 minutes on average. During the optimization, 146 successful order swap moves are made. Without the internal timetabling module, only 5 swap moves are accepted. This means that from all tested swaps, only 5 candidates are feasible and improve the current solution. If conflicts can be solved using the internal timetabling module but no potential check occurs, the number of accepted swaps rises to 11. Thus, the other 135 order swaps are selected based on their potential. Thanks to this feature, an extra 6.5% of the total spreading cost is avoided.

Using the above parameter settings, the system that results from the iterative procedure of routing and timetabling is evaluated using the simulation model of Section 3.3.4. The results are summarized in Tables 6.2-6.5. These tables extend Tables 5.7-5.10. Column *reference* corresponds to the initial situation that is used as reference and column *routing* represents the system given by the optimal routing solution. A quick look at these tables suffices to see the substantial improvement of the results in the *timetabling* column. Except for the standard deviations of the robustness and the passengers' delays in Tables 6.3 and 6.4, all performance indicators and their standard deviations are significantly better than in the other columns. Although the spreading cost decreased to 34.7%, the improvement in robustness lies between 2.6 and 2.9% based on the RWTT[16] or 4.6 to 7.2% for the stochastic component (Rob_2). Larger differences are found for the delays. Considering the propagation of delays in the stations or on the grids, see Table 6.6, a reduction is achieved everywhere. Another remarkable result is the decrease by about one third in the percentage of extra delayed trains.

6.4 Conclusions

In this chapter, the second part of the developed methodology to improve the robustness in railway bottlenecks is presented. Based on a tabu search framework, an integrated approach that exploits the potential of changing the schedule or the train sequences is implemented. By evaluating this approach on the NSC case study, its efficiency is proven. Computational tests showed that without the feasibility restoring option and the potential check, order swaps are not attractive. This shows the usefulness of the internal timetabling phase to the optimization process.

[16] In each table, the value for Rob_1 in the timetabling column lies between 97.1% (=100-2.9) and 97.4% (=100-2.6).

Table 6.2: Simulation results for the iterative procedure of routing and timetabling for delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. This table is an extension of Table 5.7. We refer to that table for the explanation of how to read it.

	reference	routing	timetabling
spreading cost (%)	100	76.3	34.7
Rob_1 (%)	100	99.5	97.2
Rob_2 (%)	100	101.1	105.9
Rob_{stdev} (min)	0.861	0.855	0.834
pax delays (min)	1.69 (0.287)	1.67 (0.285)	1.59 (0.278)
train delays (min)	162 (23.4)	159 (23.2)	148 (21.9)
knock-on (min)	35.2 (10.1)	32.5 (9.8)	21.4 (7.2)
newly delayed (%)	8.42 (2.85)	7.54 (2.75)	5.21 (2.32)
extra delayed (%)	33.5 (5.68)	30.0 (5.52)	21.4 (4.87)
worst case (min)	298	289	261

Table 6.3: Simulation results for the iterative procedure of routing and timetabling for delay scenario $\left(E_{|T|/2}, 0, E_{|T|/2}, 0\right)$. This table is an extension of Table 5.8.

	reference	routing	timetabling
spreading cost (%)	100	76.3	34.7
Rob_1 (%)	100	99.5	97.2
Rob_2 (%)	100	101.1	106.0
Rob_{stdev} (min)	0.878	0.872	0.858
pax delays (min)	1.70 (0.293)	1.68 (0.291)	1.60 (0.286)
train delays (min)	164 (24.2)	161 (24.2)	149 (22.8)
knock-on (min)	37.1 (10.7)	34.5 (10.4)	22.9 (7.9)
newly delayed (%)	10.18 (3.12)	9.19 (3.02)	6.28 (2.58)
extra delayed (%)	34.6 (5.56)	31.2 (5.44)	22.2 (5.01)
worst case (min)	298	293	278

Table 6.4: Simulation results for the iterative procedure of routing and timetabling for delay scenario $\left(E_{3|T|/4}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. This table is an extension of Table 5.9.

	reference	routing	timetabling
spreading cost (%)	100	76.3	34.7
Rob_1 (%)	100	99.5	97.4
Rob_2 (%)	100	100.9	104.6
Rob_{stdev} (min)	1.001	1.000	0.992
pax delays (min)	2.39 (0.334)	2.37 (0.333)	2.28 (0.331)
train delays (min)	222 (26.8)	220 (26.7)	207 (25.8)
knock-on (min)	42.2 (10.4)	39.3 (10.2)	26.9 (8.0)
newly delayed (%)	5.69 (2.28)	5.20 (2.25)	3.91 (2.01)
extra delayed (%)	38.4 (5.43)	34.8 (5.29)	26.2 (5.07)
worst case (min)	369	360	347

Table 6.5: Simulation results for the iterative procedure of routing and timetabling for delay scenario $\left(P^{(0.5\hat{D})}_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. This table is an extension of Table 5.10.

	reference	routing	timetabling
spreading cost (%)	100	76.3	34.7
Rob_1 (%)	100	99.7	97.1
Rob_2 (%)	100	100.8	107.2
Rob_{stdev} (min)	0.168	0.169	0.152
pax delays (min)	1.28 (0.056)	1.27 (0.056)	1.19 (0.051)
train delays (min)	121 (4.31)	119 (4.29)	110 (3.4)
knock-on (min)	14.7 (2.64)	13.1 (2.59)	3.7 (1.4)
newly delayed (%)	6.87 (2.36)	6.16 (2.30)	3.23 (1.82)
extra delayed (%)	23.5 (3.24)	20.9 (3.21)	7.5 (2.35)
worst case (min)	135	133	121

Table 6.6: The amount of propagated delays (in minutes) in the stations and on the grids for the iterative procedure of routing and timetabling for delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. This table is an extension of Table 5.11.

	North	grid NC	Central	grid CM	Midi
reference	7.3	4.7	16.0	4.1	3.0
routing	7.2	3.4	15.9	3.1	3.0
timetabling	4.6	2.1	11.0	1.9	1.7

Chapter 7

The platforming problem

Variation in arrival times often necessitates delaying of trains (on the entry tracks) due to non-availability of platforms; these delays may also cause queuing up of trains on the tracks. [...] For an efficient train transit system it is imperative that the various operations involved with movement of trains at or near a station are streamlined. One such operation, which becomes a difficult and time consuming operation at large stations, is that of platform allocation to arriving trains.

Chakroborty et al. (2008)

In this chapter, the allocation of trains to platforms is considered. In the routing module and the timetabling module of the previous chapters, the platform assignment of all trains is assumed to be fixed. When the combination of both modules gets stuck in a local optimum, this assumption is relaxed and the impact of a platform change for a train is considered. Thus, where the algorithm of the previous chapter ends, the *platforming module* begins.

Changing the platform allocation requires four decisions: which trains should get a new platform, at which station will this happen, and which platforms are to be considered for a given train. The last decision is about the routes. Since a route is platform dependent, a new route needs to be selected out of the set of candidate routes. After these four decisions are made, the selected platform change is investigated. Similar to an order swap, platform changes incur a risk of conflicts. In order to resolve possible conflicts or explore the potential of

This chapter is based on Dewilde et al. (2013).

a platform change, a recursive call to the (internal) timetabling module is added to the platforming module in a similar way as is done for an order swap.

In the following section, the layout of the platforming module is presented. Since this module completes the algorithm, the framework of the entire algorithm is sketched in Section 7.2. After that, the computational results of the full algorithm are presented. The contribution of the platforming module, the evolution of the performance indicators along the iterations of the algorithm, and the impact of a conflict within a grid or at a station are discussed. This chapter ends with the conclusions and the answer to the second research question about how to improve the robustness of the timetable for the NSC.

7.1 The platforming module

The platforming module starts if the iterative procedure of routing and timetabling gets stuck in a local optimum. In order to escape from this local optimum with a platform reassignment, the weakest links of the system that prevent further improvement are detected. Two main weaknesses are used as criteria to add a candidate platform change to the RCL. Before discussing each of them, some details about the RCL are needed.

Each element of the RCL is a quadruple that consists of a train t, a station s, a platform p, and a candidate route r. Define p^{cur} and p^{new} as the current, respectively, new platform of train t at station s. Let $B^{s}_{t,t'}$ be the smallest time span between trains t and t' in s or at the neighboring grids of s. For the ease of notation, the parameters of the platforming module are underlined in the remainder of this section. In Section 7.3, the selected values for each parameter are discussed.

Similar to the RCL condition (6.1) of the timetabling module, the platforming module focuses on the smallest minimum time spans. Thus, only the trains t and t' of which the minimum time span is one of the smallest ones,

$$B_{t,t'} \leq \min_{\hat{t},\check{t} \in T} B_{\hat{t},\check{t}} + \underline{B^{\text{margin}}}, \tag{7.1}$$

are eligible for a platform change.

(i) The first detected weakness comes from the crisscross of routes, which is characteristic for the NSC case study. The high interaction rate of the trains causes a lot of small headway buffers with many conflicts as a consequence. By rerouting a train to another platform, some interaction with other trains can be avoided and larger minimum time spans can arise. Denote with $\Gamma^{i}_{t,s}$ the number of other trains of which the route intersects, splits, or merges with the current route of train t in the surroundings of station s. Only the trains t' for which the minimum headway buffer with train t in this region, $B^s_{t,t'}$, is smaller than $\underline{B^{\max}}$ are counted. The superscript i in $\Gamma^i_{t,s}$ is used to distinguish the trips towards and away from the station. For a through train this corresponds to each side of station s. In case of a turn around action where both the inbound and outbound route traverse the same grid, $\Gamma^i_{t,s}$ is counted separately for the inbound and the outbound route. To be considered as a candidate station for a platform change of train t, the difference between the maximum number of intersecting routes at that station and $\Gamma^i_{t,s}$ for the inbound or outbound route should be smaller than the parameter Γ^{margin},

$$\max_{\hat{t}\in T} \Gamma^i_{\hat{t},s} - \Gamma^i_{t,s} \leq \underline{\Gamma^{\text{margin}}}\,. \tag{7.2}$$

(ii) Since the blocking time at a station is quite large in case of a dwell action, a train consumes a lot of platform capacity. By reallocating a train to a less busy platform, more freedom for scheduling arises. From this point of view, the high occupation rates of platforms is considered as a second weakness of the system. Define $\Psi^1_{s,p}$ as the total occupation time of platform p in station s, and let $\Psi^2_{s,p}$ be the number of trains that use this platform. For a train to be removed from its current platform, the occupation rate of that platform should be higher than a fraction of the maximum platform occupation rate,

$$\Psi^1_{s,p^{\text{cur}}} \geq \max_{p\in P} \Psi^1_{s,p} \cdot \underline{\Psi^{\text{margin}}_1}\,. \tag{7.3}$$

To be seen as a candidate new platform, the difference between the number of trains that use the new platform and those using the current platform should not increase too much,

$$\Psi^2_{s,p^{\text{new}}} - \Psi^2_{s,p^{\text{cur}}} \leq \underline{\Psi^{\text{margin}}_2}\,. \tag{7.4}$$

The fact that the condition for the current platform is based on the occupation time, while that of a candidate new platform depends on the number of trains is due to uncertainty of the occupation time at the new platform. This occupation time depends on the inbound route. For example, when this route is shorter (longer), the occupation time becomes longer (shorter).

Since both conditions (7.2) and (7.3) relate to a different aspect of the current system, it is not because only one of them is satisfied, that a platform change is not interesting. Moreover, reassigning a train in a station to another platform is only relevant if it allows to increase some minimum time spans in the surroundings of that station. Therefore, a third condition is added that is only valid for those trains with the smallest local time spans,

$$\exists t' \in T : B^s_{t,t'} \leq \min_{\hat{t},\check{t}\in T} B^s_{\hat{t},\check{t}} + \underline{B^{\text{margin},s}}\,. \tag{7.5}$$

If next to (7.1), two of the three conditions (7.2), (7.3), and (7.5) are fulfilled, a platform change for train t in station s to any platform that satisfies (7.4) will be considered.

During the algorithm, the recent history of platform changes is stored in a tabu list. If train t gets a new platform in station s, then another platform change for t in s is tabu for a number of iterations.

The decision whether a platform is suitable as new platform depends on some extra constraints. For example, infrastructure limitations such as fixed orientations of platforms and the non-existence of a route from (to) the inbound (outbound) line determine restrictions for platform reallocations. Other examples of such restrictions are trains that are bound to use certain platforms such as non-stop trains or diesel trains in a covered station. Also cross-platform transfers can limit the number of candidate new platforms.

Based on the infrastructure, some platforms can be equally likely to be selected as new platform. For example, in Figure 7.1, a two station network with three *families* of platforms is depicted. Since a reallocation to another platform within the same family has no impact on the routing through the grids, only the one with the smallest occupation rate, $\Psi^1_{s,p}$, will be considered as destination of the platform change.

When all promising (t, s, p) combinations are selected, the set of routes that are not route dominated is matched with all the triplets to obtain all RCL candidates. Next to rerouting a train to one of the candidate platforms, new routes to the current platform are also selected. This gives RCL candidates of

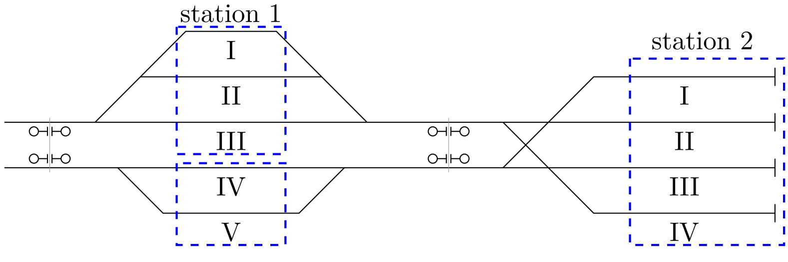


Figure 7.1: Family of platforms. In this two station network, there are three families of platforms. The platforms of station 1 can be divided in two families; platforms I-III form a family, just like IV and V. For a train approaching station 1, no distinction between the platforms of the same family can be made based on the infrastructure. Since station 2 is a terminal station, trains reverse there. As a consequence, the four platforms are part of one and the same family.

the form $(t, s, p^{\text{cur}}, r^{\text{new}})$. As a result, incompatible routes can be selected and the algorithm tries to make them compatible. Since no conflicts are allowed in the routing module, this feature allows the evaluation of routes that are unlikely to be selected otherwise.

The selection of routes ends with a reduction of the set of candidate routes. Using the routing information, the current and new number of intersecting routes can be compared. If this number grows for a certain route, the chance of improvement is estimated to be quite low and therefore, this route will not be considered.

Once the RCL is built, each candidate is considered one by one. To solve possible conflicts or to estimate the potential of a platform change, the internal timetable module is applied. At the end, only conflict-free solutions are accepted. When the internal timetabling module for a particular platform change finds a new globally best solution, only the remaining RCL elements for the same train and at the same station are considered before the platforming module ends. If no improved solution is found, the platforming module ends by returning its best solution. Thus at the end, a single platform change or rerouting action is made.

7.2 Framework of the algorithm

With the platforming module, the last component of the developed algorithm is explained. An overview of the entire procedure is repeated in Algorithm 7.1 and Figure 7.2 is used to visualize the interaction between the modules. In the first step, the TRP is solved with the routing module from Chapter 5. After that, improvements to the timetable are tested in the timetabling module. In order to enlarge the time span between two trains (represented by the spaces between the trains in Figure 7.2), timetable shifts or order swaps are performed. If the methodology from Chapter 6 succeeded in reducing the spreading cost,

Algorithm 7.1 Framework of the developed algorithm

```
input: infrastructure data and reference timetable
while number of consecutive non-improving iterations ≤ iter^max do
  solve the train routing problem (TRP)
  apply tabu search (timetabling module)
  if timetabling did not yield improvements
    then start the platforming module
```

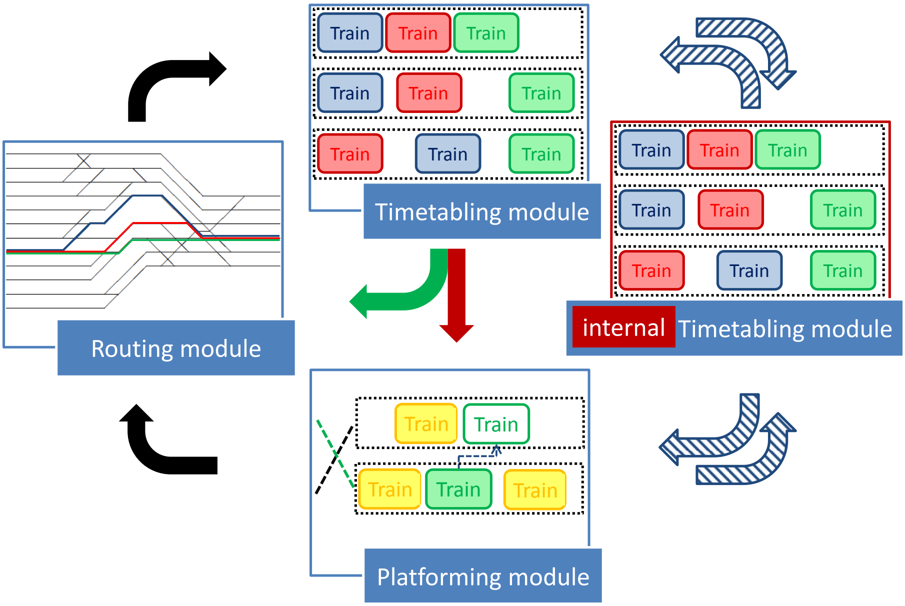


Figure 7.2: Visualization of the interaction between the modules of the algorithm.

the routing of the trains is considered again (leftward arrow in Figure 7.2). Otherwise, the downward arrow is followed and the potential of platform changes is evaluated before the routing module is called. In the platforming module, only one platform change is made. The internal timetabling module to restore feasibility or test the potential is used as part of an order swap within the timetabling module or after a platform change. The entire algorithm is stopped when $iter^{\max}$ consecutive, non-improving iterations of routing, timetabling, and platforming are made. In this case, we consider the corresponding system to be optimized for the considered network.

7.3 Results

In Table 7.1, an overview of the parameters of the platforming module is given. Next to the upper bound of relevant time spans, $B^{\max}$, the two other minimum time span margins, B^{margin} and $B^{\text{margin},s}$, that are used for inequalities (7.1) and (7.5) are given. The first one limits the width of the RCL to 2 minutes, the second one ensures that the platforming train is one of the trains with the smallest time spans in the surroundings of that station. The difference

Table 7.1: Parameter values for the platforming module.

Parameter	Value	Parameter	Value (iter)
$B^{\max}$	15 min	Ψ_1^{margin}	0.95
B^{margin}	2 min	Ψ_2^{margin}	2
$B^{\text{margin},s}$	0.5 min	platform-tabu tenure	2
Γ^{margin}	2	$iter^{\max}$	10

with the maximum number of intersecting routes of the inbound or outbound route, Γ^{margin}, is set to 2. The allowed deviations of the maximal platform occupation rate is 5% ($\Psi_1^{\text{margin}} = 0.95$) in condition (7.3) or 2 trains (Ψ_2^{margin}) in condition (7.4). Finally, once a platform is changed in a certain station, it should remain fixed for the following two calls to the platforming module (platform-tabu tenure).

The primary criterion to use these parameter values was the size of the RCL. On the one hand, since the total set of candidate platform changes is a function of the number of trains, the number of stations, the number of platforms, and the number of alternative routes, strict rules to reduce this set were needed. On the other hand, the set of candidates cannot be too small since otherwise no promising, non-tabu candidates remain after a few iterations. Based on this criterion, a valid range is determined for each parameter. In order to obtain the exact parameter values, the spreading cost of the final solutions that are obtained using the remaining possibilities are compared and the most promising settings are selected.

The stopping criterion for the entire algorithm, $iter^{\max}$, is set to 10 consecutive, non-improving iterations. In some cases, a marginal improvement can be found by increasing this value. This is the influence from the random factor within the shift neighborhood. However, since this occurred only rarely during the testing of the algorithm, the value 10 is selected.

7.3.1 Results of the algorithm

In Section 4.3, the fluctuations of the performance indicators in the course of the algorithm were mentioned. These fluctuations are now illustrated in Figure 7.3. This figure is obtained by evaluating the best solution per iteration of the entire algorithm for the NSC case study. Therefore, the simulation module is applied and delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$ is used. Remark, however, that the

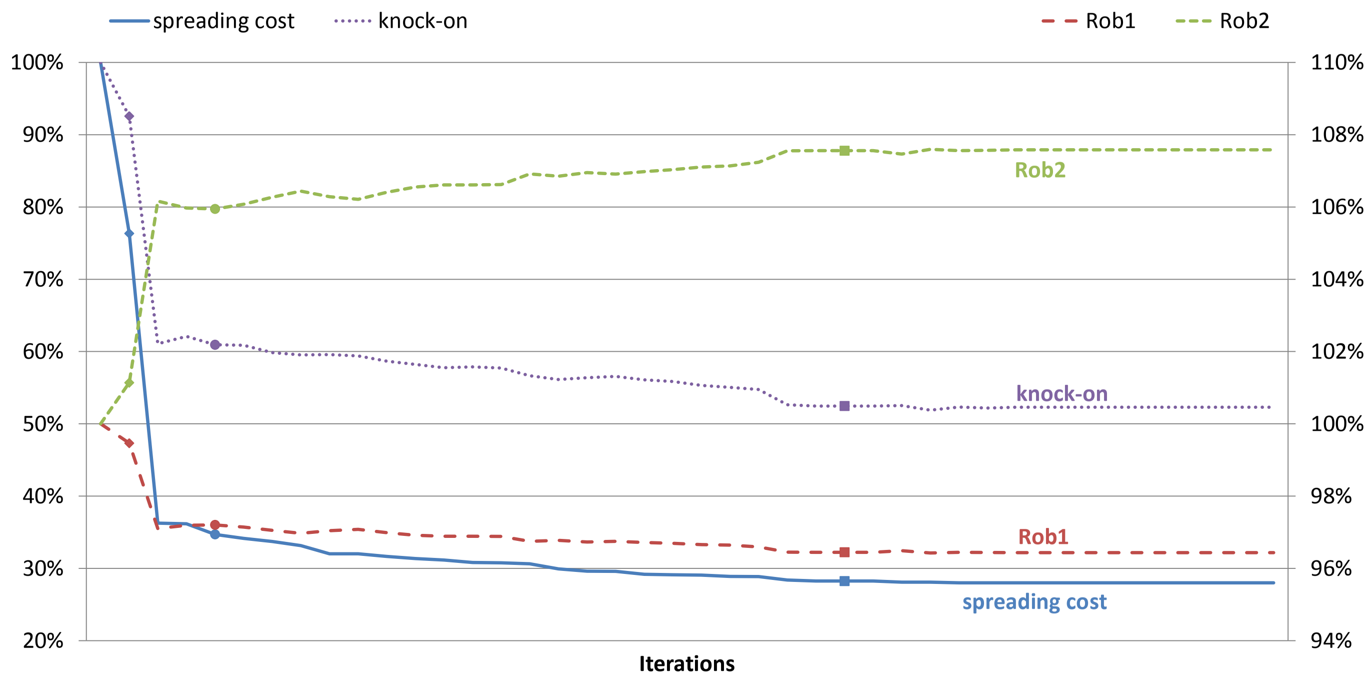


Figure 7.3: Evolution of some performance indicators in the course of the algorithm. Per iteration of the algorithm, the currently best solution is evaluated based on the objective function (*spreading cost*), the Rob_1 and Rob_2 robustness scores, and the amount of *knock-on* delays. The dashed curves for the robustness scores are scaled based on the right Y-axis, the others on the left Y-axis. The diamonds on the left hand side correspond to the optimal routing solution and the bullets indicate the simulation output of the system after the iterative procedure of routing and timetabling. The results obtained using the final solution are represented by squares.

findings are also valid for other delay scenarios and can be generalized. The results of the routing module are indicated with the filled diamonds on the curves, the bullets correspond to the solution of the iterative procedure of routing and timetabling. By adding the platforming module, the improvement process continues. Although the spreading cost goes down, the propagation of delays and the real weighted travel time (RWTT) do not always benefit from an improved spreading and show some variation along the iterations. Nevertheless, there is a clear downward (upward) trend for Rob_1 (Rob_2) and the amount of knock-on delays. Based on the importance of the different performance indicators, the overall best solution can be selected. This is the one with the lowest value for Rob_1 and the highest value for Rob_2. For the NSC case study, the best solution, which will be labeled *final*, is found after the 27$^{\text{th}}$ iteration[17]. This shows that, due to the fluctuations, the solution with the smallest spreading cost is not necessarily the best one with respect to the other performance indicators. The corresponding values of the iteration where the best solution is found, are indicated with the squares on the curves.

To obtain the solution that is selected as final solution for the NSC case study, 16 calls to the platforming module took place on a total of 27 iterations. These 16 calls all resulted in a successful platform change. One train is platformed twice at the same station but without returning to its initial platform. Unlike in some preliminary tests, no platforming move that only consists of a new route between the original platforms is selected as best possible platforming move. The iterative procedure of routing and timetabling gets stuck in a local optimum after 3 iterations. For the other 8 iterations[18], the combination of the routing module and the timetabling module could improve the solution of the preceding platforming module. This means that the potential of 8 platform changes was not fully explored yet during the platforming module. This can be due to the impact of the routing module that can change the routes of all trains, or can be the consequence of too strict stopping criteria for the internal timetabling module. However, after a platform change is accepted (even if it is non-improving), the routing and timetabling module follow, and then the full potential is explored anyway.

By analyzing the final solution, some interesting findings can be made. Thanks to the platforming module, the (planned) platform occupation times are divided

[17]Although the evolution of the performance indicators in the course of the algorithm is more or less is similar for all delay scenarios, another solution may be selected as best one for other delay scenarios. Nevertheless, for all results in this dissertation, first, delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$ is used to select the *final* solution and then the performance of this final solution is evaluated for all (other) delay scenarios.

[18]Next to the 16 platform changes, it took 3 iterations before the platforming module was called for the first time. Since in total 27 iterations are made, 8 (=27-16-3) iterations remain.

more equally among the platforms. For example, in station Midi, the standard deviation of the platform occupation times reduces with nearly 30% compared to the reference situation. Also the number of intersecting routes is reduced considerably. Where in the reference system, there are 75 pairs of trains with intersecting routes between stations Central and Midi, only 20 intersections remain in the final solution[19]. When considering the total number of train pairs that share some infrastructure within $B^{\max}$ minutes, thus including equal platforms, the reduction amounts 33% and 35% for the network between stations North and Central, respectively, Central and Midi. At last, when considering the objective function, the spreading cost has decreased with 72% compared to the reference timetable.

7.3.2 Simulation results

From the visualization in Figure 7.3, it is clear that the platforming module helps improving the robustness of the system. The exact improvement can be found in Tables 7.2-7.5. Although only four delay scenarios are used to analyze the results, more tests have been performed but not reported since they gave comparable results. Because of the variability of the delay scenarios, it is our conviction that the reality lies somewhere in between, and thus that the findings are generalizable to all delay scenarios with small input delays. The columns *reference*, *routing*, and *timetabling* in Tables 7.2-7.5 are inherited from previous chapters. In column *final*, the results of the best solution are added[20]. The difference between the last two columns represents the impact of allowing platform changes. Except for some standard deviations in Table 7.5, the results of the final solution are significantly better than those in the timetabling column. Moreover, all obtained values of the final solution show a significant improvement compared to the reference system. For all but one entry, the best results are found in the rightmost column. The only exception is the standard deviation of both robustness scores (Rob_{stdev}) in the timetabling column of Table 7.5 which is (slightly) better than that in the final column. Compared to the reference system, the robustness' improvement lies between 3.1-3.6% and 6.1-7.7% for Rob_1 and Rob_2, respectively. Although a decrease in RWTT of 3.1% may seem modest, the improvements are more than an increased level of robustness. For example, the amount of knock-on delays has decreased with nearly 50% on average. Next to that, the reduction of passengers' delays is 7.20% on average[21].

[19] To get these numbers, only the pairs of trains that do not visit the same platform in Central or Midi and of which the minimum time span is strictly smaller than $B^{\max}$ are counted.

[20] Remark once again, that one and the same final solution is used for all simulations. This is the one that is selected as best solution using the delay scenario of Table 7.2.

[21] For example, the improvement in Table 7.2 amounts 7.46%=(1.69-1.56)/1.69.

Table 7.2: Simulation results for the full algorithm on the NSC case study for delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. This table is an extension of Table 6.2.

	reference	routing	timetabling	final
spreading cost (%)	100	76.3	34.7	**28.1**
Rob_1 (%)	100	99.5	97.2	**96.5**
Rob_2 (%)	100	101.1	105.9	**107.5**
Rob_{stdev} (min)	0.861	0.855	0.834	**0.815**
pax delays (min)	1.69 (0.287)	1.67 (0.285)	1.59 (0.278)	**1.56** (**0.272**)
train delays (min)	162 (23.4)	159 (23.2)	148 (21.9)	**145** (**21.1**)
knock-on (min)	35.2 (10.1)	32.5 (9.8)	21.4 (7.2)	**18.5** (**6.3**)
newly delayed (%)	8.42 (2.85)	7.54 (2.75)	5.21 (2.32)	**4.54** (**2.18**)
extra delayed (%)	33.5 (5.68)	30.0 (5.52)	21.4 (4.87)	**18.4** (**4.40**)
worst case (min)	298	289	261	**260**

Table 7.3: Simulation results for the full algorithm on the NSC case study for delay scenario $\left(E_{|T|/2}, 0, E_{|T|/2}, 0\right)$. This table is an extension of Table 6.3.

	reference	routing	timetabling	final
spreading cost (%)	100	76.3	34.7	**28.1**
Rob_1 (%)	100	99.5	97.2	**96.4**
Rob_2 (%)	100	101.1	106.0	**107.6**
Rob_{stdev} (min)	0.878	0.872	0.858	**0.833**
pax delays (min)	1.70 (0.293)	1.68 (0.291)	1.60 (0.286)	**1.57** (**0.278**)
train delays (min)	164 (24.2)	161 (24.2)	149 (22.8)	**147** (**22.0**)
knock-on (min)	37.1 (10.7)	34.5 (10.4)	22.9 (7.9)	**19.8** (**7.1**)
newly delayed (%)	10.18 (3.12)	9.19 (3.02)	6.28 (2.58)	**5.47** (**2.44**)
extra delayed (%)	34.6 (5.56)	31.2 (5.44)	22.2 (5.01)	**18.9** (**4.55**)
worst case (min)	298	293	278	**267**

Table 7.4: Simulation results for the full algorithm on the NSC case study for delay scenario $\left(E_{3|T|/4}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. This table is an extension of Table 6.4.

	reference	routing	timetabling	final
spreading cost (%)	100	76.3	34.7	**28.1**
Rob_1 (%)	100	99.5	97.4	**96.6**
Rob_2 (%)	100	100.9	104.6	**106.1**
Rob_{stdev} (min)	1.001	1.000	0.992	**0.964**
pax delays (min)	2.39 (0.334)	2.37 (0.333)	2.28 (0.331)	**2.25** (**0.321**)
train delays (min)	222 (26.8)	220 (26.7)	207 (25.8)	**203** (**25.0**)
knock-on (min)	42.2 (10.4)	39.3 (10.2)	26.9 (8.0)	**23.1** (**7.0**)
newly delayed (%)	5.69 (2.28)	5.20 (2.25)	3.91 (2.01)	**3.45** (**1.90**)
extra delayed (%)	38.4 (5.43)	34.8 (5.29)	26.2 (5.07)	**22.6** (**4.65**)
worst case (min)	369	360	347	**335**

Table 7.5: Simulation results for the full algorithm on the NSC case study for delay scenario $\left(P^{(0.5\hat{D})}_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. This table is an extension of Table 6.5.

	reference	routing	timetabling	final
spreading cost (%)	100	76.3	34.7	**28.1**
Rob_1 (%)	100	99.7	97.1	**96.9**
Rob_2 (%)	100	100.8	107.2	**107.7**
Rob_{stdev} (min)	0.168	0.169	**0.152**	0.154
pax delays (min)	1.28 (0.056)	1.27 (0.056)	1.19 (**0.051**)	**1.18** (**0.051**)
train delays (min)	121 (4.31)	119 (4.29)	110 (**3.4**)	**109** (**3.4**)
knock-on (min)	14.7 (2.64)	13.1 (2.59)	3.7 (**1.4**)	**3.0** (**1.4**)
newly delayed (%)	6.87 (2.36)	6.16 (2.30)	3.23 (1.82)	**2.72** (**1.69**)
extra delayed (%)	23.5 (3.24)	20.9 (3.21)	7.5 (2.35)	**6.1** (**2.18**)
worst case (min)	135	133	121	**119**

Table 7.6: The amount of propagated delays (in minutes) in the stations and on the grids for the final solution for delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. This table is an extension of Table 6.6.

	North	grid NC	Central	grid CM	Midi
reference	7.3	4.7	16.0	4.1	3.0
routing	7.2	3.4	15.9	3.1	3.0
timetabling	**4.6**	2.1	11.0	1.9	1.7
final	4.7	**1.6**	**9.7**	**1.6**	**0.9**

The difference with the improvements for the train-related performance indicators such as the propagation of delays and the percentages of newly or extra delayed trains, which all fluctuate around 50%, comes from the influence of the passenger-based weights. In Section 7.3.5, we come back to this issue. Finally, also in the worst-case scenario, a considerable reduction of total delays is achieved. Next to being more punctual, the final solution also outperforms the reference system with respect to its stability. Step by step, the algorithm reduces the standard deviation of the performance indicators and makes the system more stable.

To know more about the origin of the propagation of delays, the amount of knock-on delays per station and per grid is grouped in Table 7.6. This table is an extension of Table 6.6. The reduction in the amount of propagated delays goes from 36% in the North station to about 60-70% on the grids and in station Midi[22].

7.3.3 The impact of grid conflicts

In Section 5.2.3, the importance of investigating the routing through the grids is discussed. The impact of a single conflict within a grid or at a station is compared. For the reference system, the results indicated that externally caused grid conflicts harmed more trains or passengers than a conflict at the Central station. When the routes through the station area are optimized, the differences become somewhat smaller, but the disadvantages of a conflict in a grid remains the largest. In Tables 7.7 and 7.8, the impact of grid conflicts and station conflicts are compared for the solution of the iterative procedure of routing and timetabling, respectively, the final solution of the full algorithm. The structure of these tables is the same as in Tables 5.12 and 5.13.

In these tables, the impact on the train delays of a single conflict in the Central station is larger than that of a grid conflict for the solution of the iterative procedure and the final solution. Since this is vice versa for the reference system, we conclude that the reduction of the impact of grid conflicts is larger than the reduction of station conflicts. Initially, a single conflict of 5 minutes caused on average 5.8 minutes of knock-on delays at its worst location[23]. In the optimized system, this number is reduced to 2.7 minutes on average for a conflict at the Central station. Although station conflicts give the largest average delays, the corresponding standard deviations are smaller. This means that the impact of

[22]These values are computed by taking the ratio of the absolute improvement and the initial values. For example, for the amount of knock-on delays in the Central station: 39%=(16.0-9.7)/16.0.

[23]According to Table 5.12, the value of 5.8 minutes is the resulting amount of knock-on delays due to a conflict at grid CM_2.

Table 7.7: Comparison of the impact of a conflict within a grid or at the Central station for the solution of the *iterative procedure of routing and timetabling.*

	Central	grid NC	grid CM_1	grid CM_2
Rob_1 (%)	100	100.2	100.3	100.4
Rob_2 (%)	100	91.9	88.7	83.7
Rob_{stdev} (min)	0.062	0.079	0.095	0.101
pax delays (min)	1122 (459)	1213 (586)	1250 (708)	1306 (751)
train delays (min)	7.9 (1.7)	7.8 (1.6)	7.4 (1.8)	7.7 (1.9)
knock-on (min)	2.9 (1.7)	2.8 (1.6)	2.4 (1.8)	2.7 (1.9)
newly delayed (%)	1.87 (1.11)	2.00 (1.17)	1.67 (1.19)	1.86 (1.28)
extra delayed (%)	1.90 (1.10)	2.03 (1.17)	1.70 (1.19)	1.86 (1.28)
worst case (min)	12.0	13.1	12.0	12.0

Table 7.8: Comparison of the impact of a conflict within a grid or at the Central station for the *final solution* of the full algorithm.

	Central	grid NC	grid CM_1	grid CM_2
Rob_1 (%)	100	100.2	100.3	100.4
Rob_2 (%)	100	93.6	86.5	83.2
Rob_{stdev} (min)	0.060	0.080	0.090	0.093
pax delays (min)	1084 (447)	1154 (596)	1230 (672)	1266 (691)
train delays (min)	7.7 (1.4)	7.4 (1.3)	7.3 (1.5)	7.5 (1.6)
knock-on (min)	2.7 (1.4)	2.4 (1.3)	2.3 (1.5)	2.5 (1.6)
newly delayed (%)	1.67 (0.89)	1.71 (0.95)	1.58 (0.93)	1.71 (0.98)
extra delayed (%)	1.70 (0.90)	1.72 (0.98)	1.63 (0.99)	1.74 (1.05)
worst case (min)	11.5	11	11.9	12.1

grid conflicts varies more. For example, when regarding the worst case scenario, the worst instance is still obtained by a conflict in grid CM_2. With its 12.1 minutes, the worst case impact of a grid conflict is still larger than that of a station conflict, which amounts to 11.5 minutes. Also for the passenger-related performance indicators, the grid conflicts remain the most harmful and make the system more unstable than station conflicts. Next to that, there is a clear difference in the reduction of the standard deviation of train-related measures and passenger-related performance indicators. Where the standard deviation of the train-based numbers decreases with a factor up to 3.2 for the grid conflicts and a factor between 2.0 and 2.4 for station conflicts, lowering the variability of passenger-based results is maximally 1.6[24]. The difference between the performance indicators that are measured by trains and those based on the passenger flows is due to the number of affected passengers as discussed in Section 5.2.3.

Summarizing, from a passengers' perspective, grid conflicts are the most harmful type of conflicts. Although their impact has reduced more than that of station conflicts, they remain the most unstable. From a train perspective, a conflict in the Central station is worse than a grid conflict for the optimized system. Despite the improvements of the algorithm, the impact of station conflicts did not reduce very much. This illustrates that once the timetabling, routing, and platforming are improved, the Central station can be seen as the bottleneck of the Brussels' station area.

7.3.4 The Central station as bottleneck

In Table 7.6, there is a remarkable peak in the amount of propagated delays due to conflicts in the Central station. The delay scenario that is used to obtain these results specifies that half of the trains get a primary delay during their dwell action in the Central station. This delay scenario is selected based on the observation that, in practice, the planned dwell time of 1 minute is nearly always insufficient. In the simulation, all but one delay scenario use fixed dwell delays. Since a comparison of Tables 7.2 and 7.3 indicates that the difference of using fixed versus stochastic dwell delays is small, it is decided to work with fixed dwell delays.

Although adding these dwell delays gives a more realistic delay scenario, one can wonder how the results change if there are no dwell delays at the Central station or how the system behaves in case of dwell delays at other stations.

[24] To obtain the corresponding factors, the obtained value for the reference situation is divided by that of the final system. For example, the ratio of the value for Rob_{stdev} of 0.093 in Table 5.12 and that of its counterpart in Table 7.8, 0.060, is 1.54=0.093/0.060.

Therefore, the simulation results of three extra delay scenarios are presented in Tables 7.9-7.11. In Tables 7.9 and 7.10, no initial dwell delays are inserted in the system and half, respectively, three-quarters of the trains are delayed upon arrival. These tables are the counterpart of, respectively, Tables 7.2 and 7.4 in which the same number of trains gets an arrival delay, but also half of the trains are delayed during their stop in the Central station. The delay scenario of Table 7.11 is obtained from that of Table 7.9 by adding extended dwell times in all stations.

A comparison of the results with or without dwell delays gives that the (relative) improvement of the algorithm (column final versus reference) is rather stable and independent of the dwell delays. Without the initial disturbances in the Central station, the size of the total delays is smaller. If less trains are delayed from the start, the increase of the percentage of newly delayed trains is expected.

To obtain Table 7.11, more primary delays are inserted in the system. As a consequence, the total amount of train delays is higher than in Tables 7.2 or 7.9. However, where the increase in train delays and in knock-on delays of adding dwell delays in the Central station are equal to about 18%[25], adding delays in the outer stations increases the train delays with more than 27% while the amount of knock-on delays only increased with 9%. This means that the impact of conflicts in stations Midi and North is smaller than in the Central station[26]. Since all trains pass the three stations, this result confirms the logic that the Central station is the bottleneck station.

The same conclusion is obtained by looking at the delay propagation in the stations or on the grids. Where Table 7.6 contains the results for the delay scenario with dwell delays at the Central station only, no dwell delays are used to get Table 7.12, and in Table 7.13, there are primary delays at all stations. For Tables 7.6 and 7.12, all numbers, except for the ones in column Central, are more or less equal. In column Central, the impact of (leaving out) the primary dwell delays becomes clear. Although the amount of knock-on delays in the Central station decreases, it remains a lot higher than that of the other columns. Adding delays in the outer stations makes the system more vulnerable to delay propagation, but the increase is considerably smaller than that of the Central station.

[25] From 137 minutes of train delays for the reference system in Table 7.9 to 162 minutes in Table 7.2 is an increase of $(162 - 137)/137 = 18\%$. Also the difference in knock-on delays equals 18%, $(35.2 - 29.8)/29.8 = 18\%$. For the results in column final, more or less the same results are found.

[26] By adding delays in the two outer stations, more initial delays are inserted than by adding delays in the Central station. This explains the larger increase of train delays. Since the increase in knock-on delays is smaller, less conflicts occur or the conflicts that occur have a smaller impact. Next, where the increase of both measures was proportional for delays in the Central station, more initial delays need to be inserted in the outer stations to get the same increase in propagated delays as obtained from delays in the Central station.

Table 7.9: Simulation results for the full algorithm on the NSC case study for delay scenario $\left(E_{|T|/2}, 0, 0, 0\right)$. The results in this table can be compared with those in Tables 7.2 and 7.3 to find the impact of external dwell delays in the Central station.

	reference	routing	timetabling	final
spreading cost (%)	100	76.3	34.7	**28.1**
Rob_1 (%)	100	99.5	97.6	**96.9**
Rob_2 (%)	100	101.2	105.4	**106.9**
Rob_{stdev} (min)	0.843	0.843	0.832	**0.809**
pax delays (min)	1.54 (0.281)	1.52 (0.281)	1.45 (0.277)	**1.43** (**0.270**)
train delays (min)	137 (22.7)	135 (22.6)	125 (21.3)	**123** (**20.7**)
knock-on (min)	29.8 (9.1)	27.4 (8.9)	18.0 (6.5)	**15.3** (**5.7**)
newly delayed (%)	14.18 (3.67)	12.53 (3.55)	8.42 (2.93)	**7.15** (**2.68**)
extra delayed (%)	30.5 (5.53)	27.0 (5.33)	19.1 (4.65)	**15.9** (**4.17**)
worst case (min)	267	261	244	**241**

Table 7.10: Simulation results for the full algorithm on the NSC case study for delay scenario $\left(E_{3|T|/4}, 0, 0, 0\right)$. This delay scenario is the same as in Table 7.4, except for the dwell delays in the Central station.

	reference	routing	timetabling	final
spreading cost (%)	100	76.3	34.7	**28.1**
Rob_1 (%)	100	99.5	97.7	**97.0**
Rob_2 (%)	100	100.9	104.3	**105.6**
Rob_{stdev} (min)	0.976	0.969	0.965	**0.941**
pax delays (min)	2.23 (0.325)	2.21 (0.323)	2.14 (0.322)	**2.11** (**0.314**)
train delays (min)	197 (26.2)	194 (25.9)	183 (25.0)	**180** (**24.3**)
knock-on (min)	36.3 (9.6)	33.4 (9.3)	22.4 (7.1)	**19.2** (**6.3**)
newly delayed (%)	9.85 (2.80)	8.80 (2.74)	6.45 (2.43)	**5.54** (**2.31**)
extra delayed (%)	35.6 (5.36)	31.8 (5.25)	23.5 (4.78)	**19.6** (**4.40**)
worst case (min)	334	345	321	**311**

Table 7.11: Simulation results for the full algorithm on the NSC case study for delay scenario $\left(E_{|T|/2}, P^{(0.5)}_{|T|/2}, P^{(0.5)}_{|T|/2}, P^{(0.5)}_{|T|/2}\right)$. To obtain these results, the same number of trains are delayed upon arrival as in the delay scenarios of Tables 7.2, 7.3, and 7.9. The differences come from the inserted dwell delays.

	reference	routing	timetabling	final
spreading cost (%)	100	76.3	34.7	**28.1**
Rob_1 (%)	100	99.4	97.2	**96.4**
Rob_2 (%)	100	101.1	105.6	**107.1**
Rob_{stdev} (min)	0.873	0.865	0.845	**0.829**
pax delays (min)	1.94 (0.291)	1.92 (0.288)	1.84 (0.282)	**1.81** (**0.276**)
train delays (min)	207 (24.2)	204 (23.9)	192 (22.4)	**189** (**21.7**)
knock-on (min)	38.4 (11.0)	35.6 (10.7)	23.2 (7.9)	**20.1** (**7.1**)
newly delayed (%)	2.36 (1.63)	2.11 (1.55)	1.43 (1.28)	**1.31** (**1.24**)
extra delayed (%)	35.4 (5.62)	31.8 (5.56)	22.5 (5.07)	**19.1** (**4.60**)
worst case (min)	351	351	322	**309**

Table 7.12: The amount of propagated delays (in minutes) in the stations and on the grids for the final solution for delay scenario $\left(E_{|T|/2}, 0, 0, 0\right)$.

	North	grid NC	Central	grid CM	Midi
reference	7.6	4.8	10.3	4.0	3.0
routing	7.5	3.5	10.3	3.1	2.9
timetabling	**4.7**	2.2	7.2	2.0	1.8
final	4.9	**1.7**	**6.3**	**1.6**	**0.9**

Table 7.13: The amount of propagated delays (in minutes) in the stations and on the grids for the final solution for delay scenario $\left(E_{|T|/2}, P^{(0.5)}_{|T|/2}, P^{(0.5)}_{|T|/2}, P^{(0.5)}_{|T|/2}\right)$.

	North	grid NC	Central	grid CM	Midi
reference	9.4	5.2	15.8	4.3	3.7
routing	9.2	3.8	15.8	3.3	3.6
timetabling	**5.8**	2.3	11.0	2.0	2.1
final	6.0	**1.8**	**9.6**	**1.6**	**1.1**

7.3.5 Passengers' delays and the punctuality of trains

To conclude the discussion about the results of the algorithm on the NSC case study, we take a closer look at the difference between the passenger-related performance indicators and those based on the trains. The first category comprises the robustness values and the passengers' delays. To compute the other punctuality numbers, the trains are used as unit and the passengers are ignored. Within this section, the results of Table 7.2 are used to illustrate the findings.

Consider the reference situation in Table 7.2. Comparing the results of both categories, one sees, for example, that the average delay of a train when it leaves the system equals 2 minutes[27], while the average delay per passenger is only 1.69 minutes. The difference comes from the fact that the delay of a train is measured at the time the train leaves the system, whereas the passengers' delays are measured once the passengers alight the train or, for the passengers that are still on the train, at the moment the train leaves the system. Since the delays of a train grow when this train advances in the bottleneck area, the average passengers' delays are smaller than the train delays.

In Section 7.3.1, it is discussed that the objective function during the optimization only considers trains and no passengers. As a consequence, the improvement in passenger-related performance indicators is smaller than that of train-related ones. This explains the difference in size between the reduction in passengers' delays and train delays of, respectively, 7.5% and 10.2% in Table 7.2[28].

The fact that the decrease in the amount of knock-on delays of 47.5% (from 35.2 to 18.5 minutes) is much larger than the reduction in train delays is due to the primary delays that are only included in the latter. Similarly, in contrast with the Rob_2 robustness score, the inclusion of the NTT in Rob_1 tempers the improvement in Rob_1. Although an equal reduction in travel times is the basis for both robustness scores, the inclusion of the NTT in the RWTT makes that the relative improvement in RWTT (Rob_1) is smaller than the relative improvement in RWTT_{ext} that determines Rob_2.

[27] In this table, the average amount of train delays for the reference system is 162 minutes. Since there are 80 trains in the system, each train has, on average, 2 minutes of delays.

[28] The value of 145 for the amount of train delays in column final is 10.2% smaller than the value 162 in column reference. Analogous, $7.5\% = (1.69 - 1.56)/1.69$.

7.4 Conclusions

The subject of this chapter is the platforming module. This is the last part of the developed algorithm to improve the robustness in large and complex station areas. At first, the selection process to determine which train will get a new platform at which station is explained. To pick promising platform change candidates, the selection criteria are based on the two weakest links of the system that are detected. These are the high number of intersecting routes and the unbalanced platform occupations. After the explanation of the platforming module, an overview of the entire framework of the algorithm is presented. Finally, the main part of this chapter consists of a discussion of the computational results for the case study of the Brussels' North-South connection (NSC).

Based on an analysis of the evolution of the algorithm, we conclude that our spreading objective function is an appropriate way to improve the robustness of a railway system. Moreover, the drawback of replacing our robustness functions by a substitute objective remains limited to some variation in the performance indicators. Each of the three modules contributes to the improvement process and by studying the resulting solution, we observed more balanced platform occupation times and an overall decrease in the number of intersecting routes.

Simulation is used to gain deeper insight into the quality of the newly computed timetable with its routing and platform allocation. The simulation results show that thanks to the more balanced platform occupations, the reduced interaction between the routes through the grids, and the improved spreading, the performance improves in all aspects and in all parts of the considered network. Next to the fact that less conflicts occur and fewer trains are harmed, the performance becomes more stable since the standard deviations go down. As a consequence, we conclude that the robustness of the overall system gets considerably better.

In the results, there is a remarkable reduction in the propagation of delays. About half of the initial amount of knock-on delays is avoided in the final solution of the algorithm. Another effect of the improvement process is found by comparing the impact of a grid conflict and a station conflict. Where for the reference system, a disturbance in one of the grids caused more delays than a conflict in one of the stations, the roles are somewhat reversed for the optimized system. This illustrates once more that the Central station is the bottleneck of the considered network, once the timetabling, routing, and platforming are improved. Nevertheless, this does not mean that the high interaction between the trains on the grids can be ignored. After some elaboration about the dwell delays on the simulation results, the difference between the performance indicators that are passenger and train related is discussed.

Concerning the second research question, this chapter finishes the description of and discussion about the algorithm that is developed to improve the robustness in large and complex station areas and more specifically the NSC. Next to that, the developed methodology also explains how one should deal with the limited capacity in the NSC.

Chapter 8

The applicability of the developed approach to strategic measures and new case studies

In a terminal station, the inbound and outbound trains can run in a common segment of the track in opposite directions. This leads to opposite direction conflicts which are difficult to manage as they consume unequal capacity in comparison with the other types of conflicts. [...] busy stations may be the most complex part of the network to schedule.

Rodriguez (2007)

Now that the developed algorithm to improve the robustness in complex and busy station areas is introduced, it is time to illustrate its applicability and to validate its usefulness by considering different case studies. Until now, it is assumed that strategic decisions about the network and the line planning are made beforehand and remain fixed during the improvement process. In the first section of this chapter, the impact of a number of structural measures that change the train offer or the infrastructure to further enhance the robustness are analyzed. By applying the developed algorithm to the newly created situation,

In this chapter, updated results of, among others, the case studies in Dewilde et al. (2013, 2014) are discussed.

the system explores the potential of the performed changes. Making this study provides more insight in the effect of each of these measures and illustrates the applicability of our approach to different settings. In the second part of this chapter, two new case studies are considered. At first, the network of the Brussels' NSC is extended such that it contains the incoming and outgoing lines, and afterwards, in Section 8.3, the timetable for the station area of Antwerp serves as input for our algorithm. In comparison with the NSC case study, the latter two are modeled more in detail.

8.1 What-if studies for further improvement

In cooperation with the Belgian railway infrastructure manager Infrabel, some structural measures to further enforce the robustness are investigated. Starting from the reference situation of the NSC case study from Section 4.5, some specific changes to the network or line planning are applied and the algorithm from the previous chapters is run. One of the major problems in the Brussels' area is that nearly all trains that visit Brussels dwell at one of the six platforms of the Central station. The first three measures are promising because these help to decrease the capacity usage or to increase the available capacity in the bottleneck. The fourth and last measure aims at reducing the interaction between trains by introducing corridors. In this section, the impact of these measures is evaluated and compared with each other. The idea is to make a "*what if*"-study. No claims about optimality are made nor is some sensitivity analysis performed. For example, when the number of trains in the system is reduced (measure 1), only the impact of having fewer trains is considered, and it is left in the middle whether the number and the selection of removed trains is the best possible. The measures are assessed based on their impact on the performance only. This means that no cost-benefit analysis of the necessary investments to realize the measures is done. In Engelhardt-Funke et al. (2004), the cost of investments into the existing infrastructure to increase the allowed speed is considered together with the reduction in passenger waiting time during transfers. Nevertheless, a similar cost-benefit study is beyond the scope of this dissertation.

8.1.1 Measure 1: removing some trains from the system

The first measure consists of the removal of some trains from the system. In the reference timetable, 80 trains run through the NSC between 7 and 8 AM. With this measure, the impact of reducing the train offer by 4 (measure 1_a) or 10 (1_b) trains is estimated. The removed trains are selected based on their (low) occupation rate[29]. Passengers on any of these trains are assigned to alternative trains. Similar to the principle of retime and reflow of Sels et al. (2011a, 2013a,b), this reassignment does not influence their perception of the real travel time and thus does not incur an increase in the RWTT. The reason for that comes from the assumption that a completely new schedule is constructed, which is still unknown to the passengers[30].

8.1.2 Measure 2: adapting the stopping pattern in the Central station

The second measure does not reduce the number of trains[31], but adapts the stopping pattern. More specifically, the trains that are supposed to dwell at platforms 1 and 2 in the Central station become non-stopping. Next to the two international trains that do not dwell there anyway, 26 trains become non-stopping due to this measure. Because of these non-stopping international trains, the selection of platforms 1 and 2 for this measure is logical since it decreases the heterogeneity of the stopping pattern at a platform.

After modifying the reference system, the robustness of the new system is improved. If a platform change to or from platform 1 or 2 in the Central station is applied, the stopping pattern of the corresponding train is adapted automatically such that platforms 1 and 2 remain exclusively for the non-stopping trains. This way, the algorithm can reshuffle the platform allocations to search for better solutions.

It is assumed that passengers on the trains passing platform 1 or 2 who want to leave the system at the Central station transfer at the stations North and Midi to trains that do stop at the Central station. Due to the high number of candidate connecting trains, the estimated transfer time is set equal to the minimum necessary transfer time of 5 minutes. This means that, where the through passengers on a non-stopping train gain one minute in travel time because of the avoided stop, the nominal travel time of the arriving or departing passengers rises more due to the necessary transfer.

[29]Together with Belgium's main passenger railway operator NMBS/SNCB, Infrabel drew up a list of trains that possibly could be deleted from the train offer. In measure 1_a, only the trains from the reference system that are in this list are removed. For measure 1_b, the set of canceled trains is extended with 6 peak-hour trains which are selected to equilibrate the platform occupations in the Central station.

[30]It should be noted, however, that the passenger service is affected if ones travel options are restricted or if the occupation rate of ones train grows because of the reduced train offer. Nevertheless, this is left out of this assessment. In the following chapter, changes to an already published timetable are considered and then timetable changes and removing trains from the system do influence the RWTT.

[31]Thus all 80 trains from the reference system of the NSC case study are considered here.

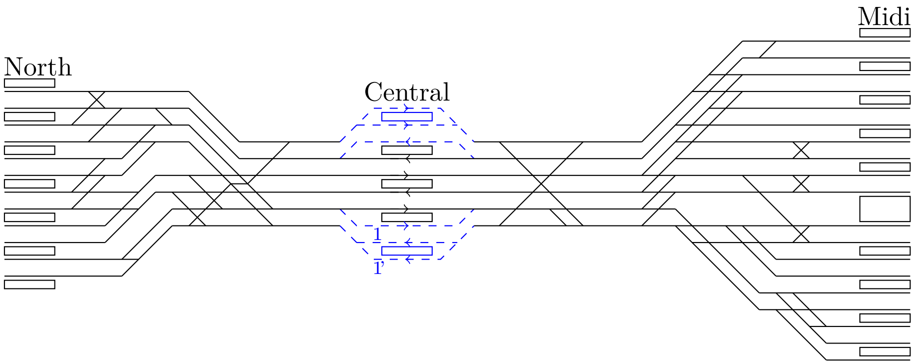


Figure 8.1: Extension of the infrastructure in the Central station as is investigated in measure 3. The newly created infrastructure is highlighted and the orientation of each platform is indicated.

8.1.3 Measure 3: extending the infrastructure in the Central station

Instead of changing the underlying line planning, the third measure focuses on the infrastructure in the Central station. The idea is to increase the number of platforms from 6 to 10 by doubling the two outer platforms at each side. This situation is illustrated in Figure 8.1. Thanks to the extended infrastructure, a train that is currently planned to dwell at platform 1, can now be assigned to platform 1 or 1'. To obtain the new system, the trains are ordered chronologically, and the odd numbered trains get assigned to the old platform, the even numbered ones to the new platform. The resulting system is then used as input for the developed algorithm in which the platforming module can improve the platform assignment.

8.1.4 Measure 4: creating a corridor model

The fourth and last measure consists of two parts that together aim at creating some kind of *corridors* through the NSC. Due to these corridors, less intersecting routes arise and fewer resources are shared by trains such that the throughput in the NSC improves. In the first part, the platforms in the stations North and Midi are (re-)oriented and the initial platform allocation is reshuffled. The second part is about limiting the routing possibilities by closing some tracks. Although the available routing options are reduced, a visualization of the impact of this measure shows that it mainly simplifies things. In Figures 8.2 and 8.3,

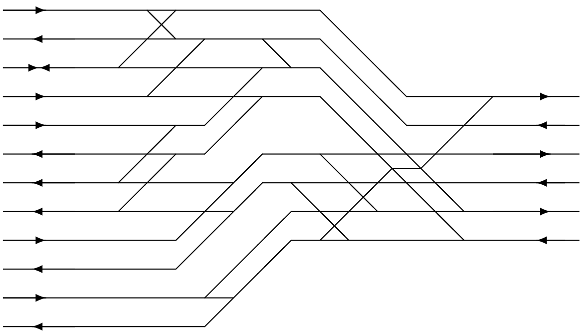


Figure 8.2: Platform orientations and infrastructure details of the reference system for the network between the North and Central station.

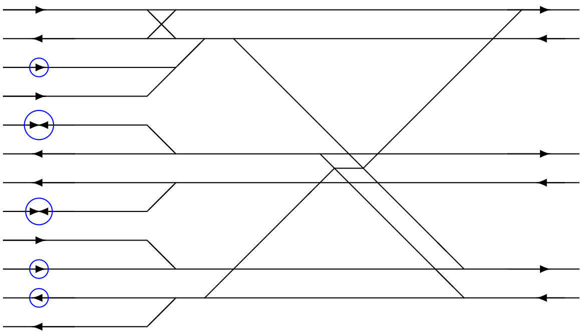


Figure 8.3: Platform orientations and infrastructure details of the corridor model of measure 4 for the network between the North and Central station. To obtain this figure, the reduced network is redrawn such that the corridors become more clear. Differences in platform orientations are circled.

the impact on the infrastructure of this measure is visualized for the network between the North and Central station. The network and the orientations of the platforms in Figure 8.2 correspond to the normal situation. To obtain Figure 8.3, the infrastructure is redrawn after some tracks are closed such that the corridors become more clear. Also the orientation of some platforms in the North station is changed. From Figure 8.3, it is clear that less intersecting routes will arise in the NSC network. To ensure that the new platform orientations are realizable, this measure should be accompanied with other infrastructure interventions outside the considered network such as the construction of fly-overs. However, this is not further considered in this dissertation.

8.1.5 Evaluation and comparison of the different measures

In order to evaluate each measure, the simulation module from Section 3.3.4 is used. Since we are now interested in the impact of the measure itself and not in the evolution of the entire algorithm, only the final outcome of the entire algorithm is considered. Next, because all delay scenarios considered in Chapter 7 lead to more or less the same conclusions, the results are reported here for delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$ only. For each measure, the values of the performance indicators are summarized in the corresponding columns of Tables 8.1-8.2 and the corresponding rows of Table 8.3. In these tables, the first column (row) equals the column (row) *final* of Table 7.2 (Table 7.6). The values in the other columns (rows) are computed as before. This means that the percentages for the spreading cost and the two robustness scores are relative to the reference system of the previous chapters. Instead of discussing how these results improve the reference system, the impact of each measure is evaluated in comparison with the improved system from Chapter 7 (*final*). Since all results are computed for improved systems, a fair comparison of the impact of each measure is obtained.

Considering the results for the systems with 76 (measure 1_a) and 70 (1_b) trains, a systematic improvement can be observed. Logically, having fewer trains reduces the capacity usage which leads to a better overall performance. The more trains that are taken out of usage, the more freedom for rescheduling arises and thus the better the results. As a consequence, the spreading cost goes down with 5 to 10% and the performance indicators that are passenger and train related such as the RWTT and the amount of knock-on delays show a considerable reduction. Taking a closer look at the amount of propagated delays in the stations and on the grids in Table 8.3, the same trend can be found. Except for station Midi, where a different platform allocation causes an increase, less knock-on delays arise in all locations. As discussed in the previous chapter, many conflicts originate due to the interaction between trains on the grids. With this measure, also the capacity usage on the grids is reduced which leads to a better performance.

Changing the stopping pattern such as in measure 2 or increasing the number of platforms (measure 3) results in a gain in the size of the total headway buffer at the Central station. Since this is the main bottleneck, these measures create more freedom for rescheduling and give a better resistance against head-tail conflicts that cause knock-on delays. Compared with the column final, the first aspect is reflected in a lower spreading cost, the second translates into better values for the train-related performance indicators. Using the robustness scores Rob_1 and Rob_2, the net impact on all the passengers of both measures can be assessed. For measure 2, the reduced Rob_1 value, in comparison with the

Table 8.1: Simulation results for measures 1-3 for delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. The first column is the *final* column of Table 7.2.

	final	measure 1_a	measure 1_b	measure 2	measure 3
spreading cost (%)	28.1	23.6	**18.5**	25.6	25.8
Rob_1 (%)	96.5	95.7	**93.2**	96.0	95.3
Rob_2 (%)	107.5	109.2	**114.5**	112.3	110.1
Rob_{stdev} (min)	0.815	0.875	0.921	0.805	**0.800**
pax delays (min)	1.56 (0.272)	1.53 (0.292)	**1.44** (0.307)	1.48 (0.268)	1.52 (**0.267**)
train delays (min)	145 (21.1)	134 (20.4)	**116** (**19.1**)	133 (20.5)	140 (20.3)
knock-on (min)	18.5 (6.3)	14.9 (5.6)	**10.6** (**4.6**)	13.3 (5.4)	13.2 (5.3)
newly delayed (%)	4.54 (2.18)	3.65 (2.02)	**2.86** (**1.80**)	4.04 (2.08)	3.58 (1.99)
extra delayed (%)	18.4 (4.40)	15.3 (4.06)	**11.1** (**3.59**)	14.7 (4.14)	15.2 (4.21)
worst case (min)	260	245	**220**	262	254

Table 8.2: Simulation results for measure 4 for delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. The first column is the *final* column of Table 7.2.

	final	measure 4_a	measure 4_b	measure 4
spreading cost (%)	28.1	36.1	**27.2**	31.3
Rob_1 (%)	96.5	96.3	96.3	**96.2**
Rob_2 (%)	107.5	107.9	107.8	**108.0**
Rob_{stdev} (min)	**0.815**	0.822	0.830	0.826
pax delays (min)	1.56 (**0.272**)	**1.55** (0.274)	1.56 (0.277)	**1.55** (0.275)
train delays (min)	**145** (**21.1**)	146 (21.6)	**145** (21.4)	**145** (21.5)
knock-on (min)	**18.5** (**6.3**)	19.3 (7.0)	19.0 (6.5)	**18.5** (6.5)
newly delayed (%)	4.54 (**2.18**)	4.67 (2.22)	4.52 (2.19)	**4.38** (2.19)
extra delayed (%)	**18.4** (**4.40**)	19.7 (4.81)	19.0 (4.51)	19.6 (4.80)
worst case (min)	**260**	308	270	284

Table 8.3: The amount of propagated delays (in minutes) in the stations and on the grids for each of the measures. The results are obtained using delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$.

	North	grid NC	Central	grid CM	Midi
final	4.7	1.6	9.7	1.6	**0.9**
measure 1_a	3.4	1.3	7.8	1.3	1.1
measure 1_b	**2.1**	**1.0**	5.3	**1.0**	1.2
measure 2	3.0	1.3	5.8	1.6	1.6
measure 3	3.2	1.9	**5.0**	1.8	1.3
measure 4_a	5.6	2.1	8.1	1.8	1.7
measure 4_b	4.8	2.1	9.8	1.3	1.1
measure 4	4.3	1.8	9.4	1.5	1.5

column final, indicates that the reduction in delay cost is larger than the growth in nominal travel time due to the necessary transfer for some passengers. Since this transfer time is not included in Rob_2, the large gain there confirms this. The difference in nominal travel time is also the reason why Rob_1 is better for measure 3 than for measure 2 while this is vice versa for the stochastic part (Rob_2).

Next to the significant improvements for the amount of delays and the percentages of newly and extra delayed trains, also the standard deviations become lower for most performance indicators. This illustrates that reducing the capacity usage or increasing the available capacity of the bottleneck has a positive impact. Thus, the large improvements for the Central station in Table 8.3 are not very surprising. Since measure 3 affects 4 platforms of the Central station and measure 2 only 2, the larger decrease in the amount of propagated delays in the Central station for measure 3 is expected. However, due to this measure, the critical block section for some pairs of trains shifts to the grids which clarifies the increase in delays there.

Corridor model

The fourth measure neither increases the available capacity nor decreases the capacity usage. On the contrary, by closing some tracks, the available capacity is reduced. To get a full understanding of the effect of this measure, the impact of each part is studied separately. In measure 4_a, the reduced infrastructure is considered with the platform allocations of the reference system. Measure 4_b is used to evaluate the new platform orientations on the current infrastructure. The results for the combination of the two, the so-called corridor model, are summarized under measure 4.

When considering the results for measure 4_a in Table 8.2, the impact of the infrastructure limitation is found. At first, one sees the increased spreading cost. Next, there are the differences for the passenger-related performance indicators which are, statistically seen, not significant. The worsening for the train performance indicators, however, is significant. The reduced minimum time spans result in more train delays and more harmed trains. Also the standard deviations and the worst case performance grow. Details about the knock-on delays (Table 8.3) reveal that more conflicts occur on the grids and in the outer stations. This is due to an increased interaction between trains on the grids.

The impact of redistributing the trains to other platforms is partly undone by the platforming module in the algorithm. The platform orientations, however, are respected at all times. Because of the platforming module, the differences between columns final and measure 4_b are not large. Except for the standard deviation and the average values of the knock-on delays and percentage of extra delayed trains, the other results are not significantly different from each other.

Doing the same exercise on the reduced infrastructure gives different results. In comparison with the final solution, the train delays decrease for measure 4. This shows that when combining the two actions, the negative effect of each of these disappears and becomes an advantage. Except for the total amount of delays and the amount of knock-on delays, where no significant differences are measured, and the percentage of extra delayed trains that remains worse, the other performance indicators become better. Thus, by changing the routes outside the NSC, less switches are needed to reach the right platform in the Central station and the interaction between trains reduces. Comparing the spreading costs in Table 8.2 gives that the infrastructure reduction is responsible for the increased spreading cost of measure 4.

A further comparison of the results for measure 4 shows that the platform reallocation has a positive impact on the system with the reduced infrastructure; comparing the results for measure 4_a with those in column measure 4, one gets significantly better results for the latter regarding the amount of train delays and the percentage of newly delayed trains. At the stations North and Midi, the amount of propagated delays lessens and this also goes along with fewer delays on the grids. By sending a train to a platform in North or Midi that is *closer* to its platform in the Central station, fewer switches are needed than before the platform reallocation. This explains these results.

Summarizing, the best results are found by the measure that removes some trains from the system. When the capacity consumption is reduced in all parts of the network, the flow through the bottleneck improves and the negative effect of the interaction between trains on the grids lessens. The measures that mainly focus on the Central station, measures 2 and 3, also improve the current situation and shorten the RWTT but to a smaller extent than measure 1. The fourth measure, which does not focus on the Central station, but more on the interaction between trains, shows that grouping the platforms and evolving towards corridors can also help to bring the performance to a higher level.

From another point of view, we can conclude that the results of these experiments illustrate that the developed algorithm perfectly serves the purpose of dealing with changes in strategic decisions and evaluating them.

8.2 The entire Brussels' area

Until now, the network of the North-South connection (NSC) at the heart of the Brussels' station area is used as case study. In this section, this network is extended such that it includes the beginning of the open tracks, the outer grids, and the entrances to the shunt yards. Doing so, the full interaction between the trains in the entire station area of Brussels is captured. An overview of the network of this section's case study is given in Figure 8.4. In this figure, abstraction is made of the detailed infrastructure. Nevertheless, the fact that the limited infrastructure makes the NSC the bottleneck is obvious. What is less clear, is the complexity of the grids outside the NSC. To connect all platforms of station Midi with all incoming and outgoing lines, a complex grid that allows numerous routing options is located right next to the station. An illustration of this grid, as well as the grid towards the Central station is given in Figure 8.5. Notice that for the NSC case study, only the 19 through platforms of station Midi are considered, while now, the dead-ending platforms that are oriented away from the NSC are also taken into account.

Next to a larger network, there are several other improvements in the case study of the entire Brussels' area compared to the NSC case study. First of all, the new case study is modeled more in detail. The exact location of the signals determine the block sections and their mutual distances. In comparison with the sectors with fixed traversing times that were used before, each train type has its own speed limitations and each route its own duration. As a consequence, differences in arrival time of selecting alternative routes can arise. When a train reaches its end station within the considered network, reutilization follows or the train continues to a shunt yard. These actions, just as split or merging actions, are considered in the new case study. Often, dwell time supplements come along with any of these actions.

One of the main causes of knock-on delays is the high interaction rate between trains in the NSC. When considering the entire Brussels' station area, many more intersecting routes occur. Often, two trains share resources at both sides of the station area. This puts more constraints on the timetable such that less freedom for improvement remains. Since the time a train needs for crossing the network has grown, a disturbance that occurs in an early phase of a train's trip through the network tends to cause more hinder to other trains and affects more passengers. Thus, if one succeeds in avoiding (early) conflicts, a larger impact than in the NSC case study is expected, if enough freedom for improvement remains.

Figure 8.4: A schematic overview of the entire station area of Brussels[32]. The NSC is represented by the three parallel tracks in the middle of the figure. The tracks to the top right are in the direction of Antwerp.

[32]Source: http://www.infrabel.be/sites/default/files/documents/ns_c-01-map-net-10459-01_1.pdf, consulted in September 2014

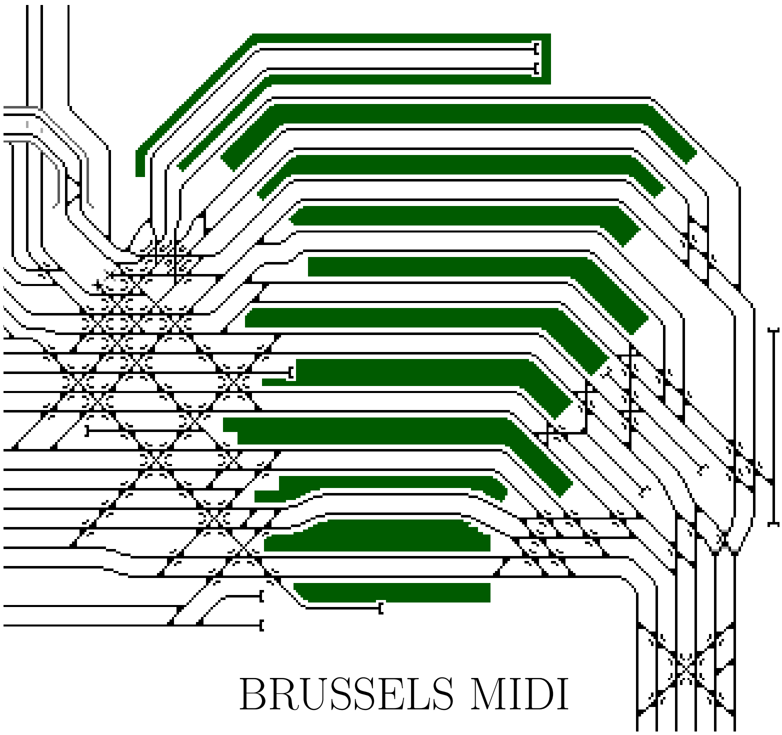


Figure 8.5: The complexity of the grids at Brussels Midi[33]. The tracks at the right hand side of the station lead to the Central station and are included in the NSC case study. The more complex grid at the left hand side is new compared to that case study.

[33]Source: http://www.sporenplan.nl/, consulted in September 2014

Next to the stations Brussels Midi, Central, and North, a fourth station, Schaarbeek, is included in the larger network. In this station, many (peak hour) trains start their journey, but also many other trains pass the station without stopping. The timetable that is used as input timetable (reference) consists of 84 trains that pass the station area between 7 and 8 AM. In the original version of this timetable, there were more trains but often the same rolling stock unit was counted twice, for example, after a reutilization. To simplify the computations, some of these actions are grouped as actions of one and the same train. For example, a train with Brussels Midi as final destination that continues its journey towards a new destination (without a turn around) is now seen as one train instead of two different ones. Doing so, the late arrival of the first train automatically influences the departure of the second train, as is the case in practice. Since reversing trains use the same infrastructure twice, they are not united as one, but strict rules about the order of events are applied.

8.2.1 Results

When the developed algorithm is applied to the case study of the entire Brussels' area, less iterations are made than for the NSC case study. Due to the reduced freedom, only 9 platform changes are made before the most robust solution is found. Next to five trains that get a new platform in station Midi, one train is routed along another platform in the Central station, one in the North station, and two platform changes are applied in the station of Schaarbeek.

Thanks to the effectiveness of the preprocessing, the Kaufman-TRP model is able to solve the TRP for the considerably larger instances in two seconds on average. Next to facing larger instances, some extra freedom is inserted in the routing module by allowing trains to enter the shunt yard from another, neighboring track. As illustrated in the figure about the different steps in railway planning, Figure 2.1, is the rolling stock schedule, which includes planning the shunt actions, composed after the timetable creation process. Together with the assumption that conflicts outside the considered network are not impossible to solve, it is assumed that the system can cope with trains that enter the shunt yard from their currently assigned track or a neighboring track. In total, an alternative track is selected for 11 of the 32 trains that come from or head to the shunt yards.

Where the impact of an improved routing solution is limited in the NSC case study, a reduction of the RWTT of 4.1% (Rob_1) is achieved now[34]. This corresponds to nearly 20% less propagated delays what shows that the routing through the network has a large impact on the performance. Applying the

[34] The simulation output for the optimal routing solution is not included in a result table.

timetabling module and the platforming module ameliorates the results even more. The final results for the case study of the entire Brussels' area are presented in Table 8.4. Compared with the NSC results, the reduction in spreading cost of 66.2% is a bit smaller, but for the same delay scenario as in Table 7.2, the (positive) impact on the passenger-based performance indicators and the train delays is larger. With respect to the reference system, the robustness has improved with almost 10% based on the Rob_1 measure and more than 17% when only considering the stochastic part (Rob_2). The total amount of knock-on delays has decreased from 219 minutes to 141 minutes, which is a reduction of 35.7%, and also the percentages of extra or newly delayed trains went down. Next to shorter real travel times and less delays, the simulation results for the final solution also show reduced standard deviations. Except for the percentages of newly and extra delayed trains for which the variability remains more or less equal, the standard deviations go down with 23.8 to 34.1%.

To end this section, some details about where the conflicts occur and how these impact the propagation of delays is presented in Table 8.5. In order to obtain this table, the entire Brussels' station area is divided in 7 zones: the 5 zones of

Table 8.4: Simulation results for the entire Brussels' station area for delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$.

	reference	final
spreading cost (%)	100	33.8
Rob_1 (%)	100	90.6
Rob_2 (%)	100	117.1
Rob_{stdev} (min)	2.53	1.93
pax delays (min)	3.91 (0.85)	3.22 (0.64)
train delays (min)	298 (70.4)	237 (50.7)
knock-on (min)	219 (74.2)	141 (48.8)
newly delayed (%)	15.7 (3.32)	12.4 (3.33)
extra delayed (%)	47.8 (5.94)	36.6 (5.93)
worst case (min)	751	562

Table 8.5: The amount of propagated delays (in minutes) in the stations and on the grids of the entire Brussels' station area for delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$.

	outside N	North	grid NC	Central	grid CM	Midi	outside M
reference	21.0	19.9	9.8	22.9	13.2	20.2	53.9
final	15.0	13.2	6.2	15.7	8.8	17.0	17.6

the NSC and the two outer grids *outside N(orth)* and *outside M(idi)*. Notice that station Schaarbeek is included in the first outer grid. The structure of this table is similar to Table 7.6. The results show that the amount of knock-on delays decreases with about 30% in all zones except in station Midi and grid outside Midi. In the reference system, a lot of knock-on delays arise due to conflicts in the latter. By rerouting the trains and selecting some alternative entrances to the shunt yards, the amount of propagated delays is already more than halved. In the end, a reduction of about two third is achieved there.

8.2.2 Impact of the weights that represent the value of travel time

Following the discussion about the weights of the RWTT in Chapter 3, the computational experiments are repeated with all weights equal to 1[35]. Therefore, all nonzero weights in Table 3.1 are set to 1. In this case, only the real travel time is counted and thus there is no difference in the weight of delays and that of the nominal travel time. There is also no distinction between the usage of a running time supplement (RTS) since both, useful and non-useful supplements extend the real travel time. Since the values of travel time are only used to compute the robustness scores and its standard deviation, only these results change in comparison with the results in Table 8.4. Instead of a reduction in RWTT of 9.4% $(Rob_1 = 90.6\%)$, the average real travel time decreases from 14.2 minutes to 13.5 minutes, which is a reduction of 4.9%. Concerning the stochastic part of the real travel time, the improvement of 17.1% for Rob_2 becomes 16.6% if all weights are equalized. Thus, although the size of the improvement reduces, the improvement remains significant. The same trend is found when repeating the computations for the NSC case study. In this case, the Rob_1 score increases from 96.5% to 98.3%.

8.3 Antwerp

The last case study to illustrate the applicability of the developed methodology is based on the station area of Antwerp. This station area is significantly different from that of Brussels. The network of Antwerp is depicted in Figure 8.6. On the left hand side of this figure, an overview of the entire area is shown and more details are presented at the right hand side. Note that tunnels are represented

[35] As indicated in Section 3.5.2, a full sensitivity analysis of the impact of the weights is out of scope of this dissertation. With this section, some results to gain some insight in this sensitivity are presented without claiming completeness.

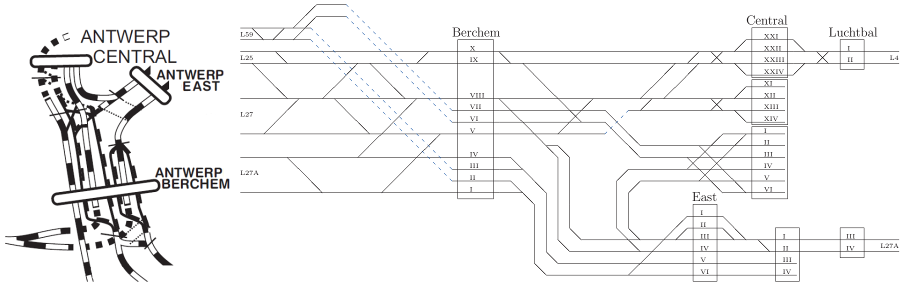


Figure 8.6: An overview of the station area of Antwerp[36]. The dashed lines represent tunnels.

by the dashed lines. In the station area of Antwerp, the two major stations, Berchem and Central, are connected through three corridors allowing trains to arrive at three different levels in the Central station. The high capacity usage by a heterogeneous fleet of trains with international trains that mix up with slow and fast local trains and freight trains, makes the station area an interesting case study for our purposes. Other challenges are that all but four platforms in the Central station are dead-ending such that many ordering constraints are needed to model the reutilizations. Moreover, as is mentioned in the quotation at the beginning of this chapter, terminal stations are characterized by the unequal capacity consumption that comes from conflicts between trains in opposite directions that request the same infrastructure simultaneously (Rodriguez 2007).

Unlike in Brussels, the specific layout of the infrastructure in Antwerp limits the routing and platforming possibilities considerably. For example, for the platforms in the station Antwerp Berchem, there is a one to one relation with the lines on the outside of the network. This means that the origin and destination of a train normally determine the platform. Next to the fact that the amount of switches is rather limited (for sure compared to the other case studies), this diminishes the impact of the routing module and restrains the options for the platforming module. Also the fact that a turn around requires a large free time slot at a platform does not help the platforming module. Nevertheless, when applying the algorithm to reduce the spreading cost, the impact on the robustness and the other performance indicators show that a significantly better system can be obtained.

[36]Source: http://www.infrabel.be/sites/default/files/documents/ns_c-01-map-net-10459-01_1.pdf, consulted in September 2014

8.3.1 Results

The results for the case study of the station area of Antwerp are presented in Table 8.6. A former timetable from practice with 90 trains in a time span of two hours is used as input. By considering two consecutive periods, modeling turn around actions that exceed the hour becomes easier. As a consequence, the values in Table 8.6 are totalled for all trains within the considered two hour range. The results in Table 8.6 are obtained using delay scenario $\left(E_{|T|/2}, P^{(0.5)}_{|T|/2}, P^{(0.5)}_{|T|/2}\right)$. This means that half of the trains enter the system with a stochastic delay and half of the trains experience extended dwell times at stations Berchem and Central. In both cases, the dwell delays are deterministic and equal half a minute.

The decrease in spreading cost to 27.6% of the initial amount resulted in a considerably reduced RWTT. The improvement in the stochastic part of the robustness of 14.2% for Rob_2 is of the same order of magnitude as the reduction in train delays of 18.8%[37]. Next, the evolution of the amount of knock-on delays and the percentages of newly and extra delayed trains show that less conflicts occur with less propagated delays as a consequence. This is also reflected in the smaller standard deviations for all performance indicators which shows that the final solution is significantly more stable than the reference system.

When the station area of Antwerp is divided in zones as in Figure 8.7, the delay propagation per zone can be studied. At each end of the considered network, there is a zone containing the incoming lines and the grid: *outside B(erchem)*, *outside L(uchtbal)*, and *outside E(ast)*. The other three zones correspond to the stations and the network between these stations (grid BC). The total amount of knock-on delays per zone that is gathered by all trains in the considered period of two hours is summarized in Table 8.7. The difference in the size of the delays is remarkable. On the one hand, the one to one relation between the platforms and the incoming and outgoing lines leads to very low knock-on delays in the zones outside Berchem and outside Luchtbal. On the other hand, there is the influence of the terminal station. If a platform is still used for a turn around action, an arriving train has to wait and gets delayed. This creates a spill back effect and is noticeable in Table 8.7. Since the platform occupation times are much larger in terminal stations, there is a larger tread on facing an occupied platform and the average remaining occupation time that determines the size of the knock-on delays is larger than in through stations. Note that no real-time rerouting or replatforming interventions to avoid such situations are made in the simulation module. Together with the fact that the departure event can

[37]The total amount of delays for all trains within the 2 hour range goes down from 352 minutes (column *reference* in Table 8.6) to 286 minutes which is 81.2% of the reference amount.

Table 8.6: Simulation results for the case study of Antwerp for delay scenario $\left(E_{|T|/2}, P^{(0.5)}_{|T|/2}, P^{(0.5)}_{|T|/2}\right)$. The values in the table are totalled for all trains within the considered two hour range.

	reference	final
spreading cost (%)	100	27.6
Rob_1 (%)	100	90.6
Rob_2 (%)	100	114.2
Rob_{stdev} (min)	2.06	1.86
pax delays (min)	3.65 (0.69)	3.13 (0.62)
train delays (min)	352 (60.6)	286 (47.1)
knock-on (min)	160 (49.3)	92 (31.3)
newly delayed (%)	13.8 (3.15)	9.5 (2.89)
extra delayed (%)	47.8 (6.16)	33.6 (5.79)
worst case (min)	702	565

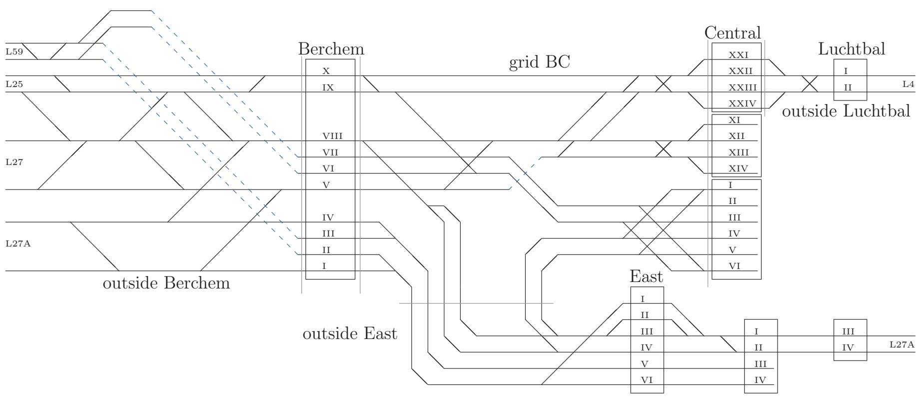


Figure 8.7: The grids within the station area of Antwerp. Since the tracks towards station East are hardly used by the trains in the timetable, the grid outside East is not included in the results of Table 8.7.

Table 8.7: The amount of propagated delays (in minutes) in the stations and on the grids of the station area of Antwerp for delay scenario $\left(E_{|T|/2}, P^{(0.5)}_{|T|/2}, P^{(0.5)}_{|T|/2}\right)$.

	outside B	Berchem	grid BC	Central	outside L
reference	3.4	45.6	38.1	46.3	5.4
final	0.6	24.9	19.1	31.1	4.3

only take place after the turn around of the arriving train is completed, it is clear that a system with multiple turn around actions is more vulnerable to knock-on delays. In our algorithm, however, no specific measures to reduce the interaction due to reutilizations is considered. Nevertheless, compared to the reference situation, the reductions in the amount of propagated delays range from 32.8% in the Central station to 50% on the grid between Berchem and Central. Although we are convinced that there is potential for further improvement by focusing on the turn around actions during the optimization and/or by altering the reutilizations, for example, by assigning a train to another line after it has terminated, this is not explicitly considered in the algorithm.

Comparing the results for Brussels (Table 8.4) and Antwerp, one sees the same trends. There is a huge decrease in the spreading objective function value, and the RWTT shortens twice with about 10%. Also the reduction of the train delays or passengers' delays, and the propagation of delays is more or less similar in both case studies. Together with the fact that the case study of Antwerp is completely different from that of Brussels, this supports the conclusion that the developed algorithm is suitable to improve the robustness in densely used, large and complex railway stations.

8.3.2 Validation by commercial simulation tool

This conclusion is supported by a validation study using one of the Belgian railway infrastructure manager's commercial (microscopic) simulation tools LUKS (Janecek et al. 2010; VIA-CON 2013). Tourwé (2014) presents the results of this study. The output is based on 100 simulation runs. Each of these runs is initiated by a set of primary delays that are calibrated to fit historical data. Conflicts are handled by dispatching rules that imitate the traffic control actions. For this validation, the study area of Antwerp is extended such that it contains an extra stopping station for each train. The reference timetable is compared with the final timetable that is computed with the developed algorithm.

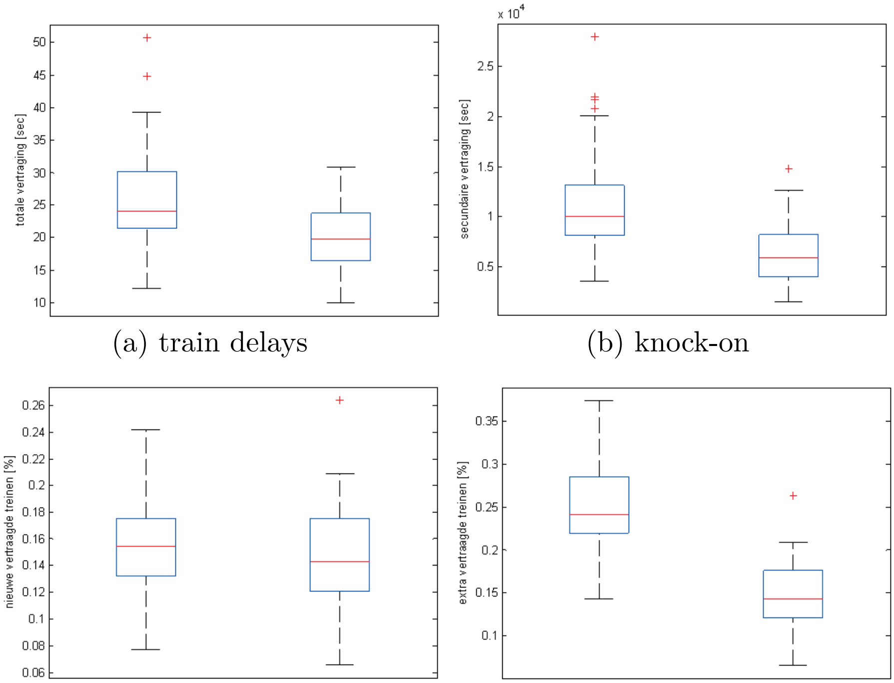
The first important result concerns the feasibility of the input timetable in case no primary delays are inserted in the system. Due to the extension of the network and some differences in modeling approach, three conflicts are detected for the final solution. These conflicts are (i) located outside the network of Figure 8.6 and potentially caused by assumptions about the usage of RTS in LUKS, (ii) located outside the station area and too small to correspond to a real conflict, or (iii) located within the considered network but due to a difference in route choices[38]. As a consequence, it can be concluded that extending the network did not give (real) problems with respect to the feasibility of the improved timetable.

Second, the measured improvements to the system are of the same order of magnitude as the results we present. For example, the decrease in train delays and knock-on delays is estimated at about 17.9% and 40.9%, respectively, while we obtained 18.8% and 42.3% using our self-developed simulation tool. Tourwé (2014) presents the results graphically in the form of box plots. In Figure 8.8, the most important results are shown. Notice that LUKS only considers trains and ignores passengers and passenger actions such that the passenger-based performance indicators that are used in this dissertation could not be compared.

In part (a) of Figure 8.8, the box plots of the amount of train delays is shown. Since the box at the left (reference system) lies clearly higher than the box at the right (final solution), fewer train delays arise in the optimized system. The same holds for the box plots in part (b). In this part of the figure, the results for the cumulative amount of knock-on delays are plotted. From this plot, it is clear that the range of the data points (including outliers) is smaller for the final solution. Part (c) contains the results for the percentage of newly delayed trains. Here, the differences between both box plots are not as large as before. Due to differences in the delay scenarios that are used by our simulation tool and by LUKS, the average improvement and the difference in standard deviations do not match. For the percentage of extra delayed trains in part (d), however, the impact of the initial delay scenario has faded and a large improvement becomes visible. We refer to Tourwé (2014) for more details. Next to the graphical comparison, Tourwé (2014) also tested the significance of the improvements. Using a significance level of 5% like is done in this dissertation, the improvement achieved for all performance indicators and thus also the percentage of newly delayed trains is found to be significant. Together with the other results of this study, this concludes the validation of our findings.

For the station area of Brussels, no similar validation study is performed yet.

[38] In LUKS, only predetermined routes are used and no alternative route choices are considered. Due to the limited route choices in the station area of Antwerp, this forms no real drawback for this comparison. Nevertheless, the single conflict that was detected within the network of Figure 8.6 was due to an alternative route that is selected in the final solution of the algorithm.



(a) train delays

(b) knock-on

(c) newly delayed

(d) extra delayed

Figure 8.8: Graphical representation of the output of the commercial simulation tool LUKS. The box plots are taken from Tourwé (2014). In part (a), the box plots of the amount of train delays per train (in seconds) are drawn. The box plot at the left hand side represents the reference system, the one at the right hand side the final solution of the algorithm. Part (b) contains the results for the cumulative amount of propagated delays (in 10^4 seconds). The box plots at the bottom show the percentages of newly, respectively, extra delayed trains.

8.4 Conclusions

In this chapter, the computational results of the entire algorithm for a number of strategic measures and new case studies are presented. The effect on the robustness of measures that alter the available capacity or capacity usage is assessed for the case study of the NSC. This showed that the developed methodology can cope with changing resources or different settings. Both, lowering the ratio between the used capacity and the available capacity in the Central station and reducing the interaction between trains proved to be useful for improving the robustness. The best results are found for the combination of the two strategies. For example, when the number of trains in the system is reduced. With this study, research question 3 about the impact of structural measures on the robustness of the NSC is answered.

In the second part of this chapter, two more realistic and more representative case studies are discussed. When considering the entire Brussels' station area, the impact on the outer grids of, for example, timetable changes are taken into account. The results of the case study of the station area of Antwerp, which is completely different from that of Brussels, demonstrate that the algorithm is suited to improve the robustness in densely used, large and complex railway stations with various settings. Together with the positive results of the validation study by one of Infrabel's commercial simulation packages, the results from this chapter illustrate that the developed approach can be used for other bottlenecks. This validation is the subject of the first part of the fourth research question which is thus answered.

Chapter 9

Impact of maintenance on the railway system

Research that explicitly focus on minimizing the disturbance (of maintenance actions) to the traffic is rare, but Lake et al. (2002) falls into this category.

Forsgren et al. (2013)

To illustrate the general applicability of both the definition of robustness and the developed algorithm to improve the robustness in practice, the planning problem is considered from a different point of view in this chapter. The starting point is a published timetable that needs to be adapted due to the temporary unavailability of some resources. A planned closing of some parts of the infrastructure reduces the capacity of a railway system and makes it more vulnerable to conflicts and delay propagation. Taking the performance of the updated system into account is a new approach to rescheduling in case of a planned infrastructure unavailability. Since the timetable is in operation, changes, such as timetable shifts or cancelations, impact the passengers who need to adapt their travel behavior. In the light of passenger service, a trade-off is made between this inconvenience and the delays that occur in practice when operating according to the updated schedule. After introducing the new settings, it is shown that the presented definition of robustness and the developed methodology are useful to make this assessment.

9.1 Introduction

When a track section is blocked due to infrastructure works, a diversion and often some delays can be expected. In general, track maintenance or railway construction works are planned well in advance and, when possible, railway infrastructure managers try to schedule these interventions during periods of low or no operation. When this is not possible, the train schedule needs to be adapted. This is the situation that is considered in this chapter. Where all related research is about scheduling the maintenance itself or about the impact the maintenance has on the train planning, we go one step further and focus on the impact of the maintenance on the performance of a railway system. In this chapter, the maintenance planning is considered as given, and rerouting and rescheduling actions are performed to keep the total hindrance for the passengers as low as possible.

9.1.1 Problem description and approach

A planned unavailability of some parts of the railway infrastructure occurs due to, for example, planned track maintenance or construction works. The term *maintenance* is used to cover all these reasons. In contrast with the (relatively short) maintenance actions that are studied in Van Zante-De Fokkert et al. (2007), large maintenance (or construction) tasks are considered which last (much) longer such that interference with trains in operation is inevitable. Because maintenance is very costly and the scheduling of maintenance interventions is very difficult and subject to many constraints (Budai-Balke 2009), we consider the maintenance planning as given (and unchangeable) such that the tracks that are closed for maintenance, the so-called *track possessions*, are known well in advance. Due to these track possessions, the *original* timetable and routing becomes infeasible. For example, if a track that is normally used by some trains is closed for maintenance, new routes are needed for these trains. To distinguish conflicts between two trains (*train-train conflicts*) from *train-track possession conflicts*, the term *maintenance conflicts* is used. Similar to a (train-train) conflict-free timetable, a timetable without maintenance conflicts is said to be *maintenance-free*.

The presence of the track possessions reduces the available capacity, which is defined as the number of trains that can be scheduled in a conflict-free and maintenance-free way. To avoid the maintenance conflicts, some rerouting and rescheduling actions are required. However, adapting the schedule with the only purpose to avoid the train-track possession conflicts is rather myopic and can lead to large delays during operation. That is why the robustness of the updated train schedule is highly relevant. This is where the algorithm from Chapters 4-7 comes into play. Where the algorithm is initially designed for a network containing some large stations, the version that handles track possessions is more general and can deal with any kind of network. When making the modifications to the published timetable, the goal that is strived for is to minimize the worsening in service level for the passengers. This corresponds to (i) the condition that no train can be routed over any of the closed tracks during the maintenance period (maintenance-free), (ii) the timetable for the remaining trains should be conflict-free, (iii) the number of trains that will be taken out of operation to fit the previous condition should be as small as possible, and (iv) the *updated* schedule should be as robust as possible.

When the planned arrival or departure time of a train is shifted to another moment (earlier or later) in order to avoid a conflict with a track possession, we model this as a change in the journey time for the passengers on this train. This is in contrast with the impact of the measures of previous chapter which were aiming at a completely new schedule instead of a temporary update of a timetable that is already being used. Therefore, the impact of changes to the published timetable do play an important role here. The term *planned delays* is used to indicate the differences in journey time. As the name suggests, planned delays are known in advance since these correspond to the difference in travel time between the *original* and the *updated* timetable and are scheduled to allow for maintenance. In contrast with the planned delays, the *real* or *unplanned delays* are the commonly known delays that occur, for example, due to train-train conflicts during operation and are not known a priori.

Define Δ_{dep} (Δ_{arr}) as the difference in departure (arrival) time for a train due to the rescheduling actions. If the train departs later (earlier) in the updated timetable than in the original one, Δ_{dep} becomes larger (smaller) than 0. The same holds for Δ_{arr} in case of late or early arrivals. Using this notation, the planned delays are measured as follows

$$\text{planned delays} = \begin{cases} \max\left(|\Delta_{dep}|, |\Delta_{arr}|\right) & \text{if } sgn(\Delta_{dep}) = sgn(\Delta_{arr}), \\ |\Delta_{dep}| + |\Delta_{arr}| & \text{otherwise,} \end{cases} \tag{9.1}$$

with $sgn(x)$ the sign operator which is 1 if $x \geq 0$ and -1 otherwise. An earlier departure or postponed arrival is considered as a disadvantage. When a train is scheduled to depart later or arrive earlier than in the original timetable, this can be advantageous as well as disadvantageous for the passengers (Savelberg et al. 2010; Tseng et al. 2005). For example, if it results in a longer waiting time before or after the trip by train. Therefore, all timetable deviations are considered in (9.1). The only exception is the situation where the travel time shortens. However, since the maintenance reduces the available capacity, shorter travel times cannot be obtained.

In case of a planned cancelation, passengers (temporarily) need to take an alternative train or search for an alternative mode of transport. Also in this case, the definitions of Δ_{dep} and Δ_{arr} are applicable and (9.1) is used to measure the planned cancelation delays. The planned (cancelation) delays are used to measure the inconvenience of the passengers since they have to adapt their journey to the new schedule.

9.1.2 Assumptions

This chapter should be seen as a proof of concept only, in the sense that some assumptions make the computational experiments not as realistic as those of the previous chapters. In this section, an overview of the most relevant assumptions is given.

This chapter focuses on infrastructure works that last for a number of days such that the transition between the original timetable and the updated one happens during periods of no or low operation. Moreover, the start and the duration of the resource unavailability is considered given and fixed. This means that no real-time changes are considered.

Although safety regulations imply speed limitations on tracks neighboring the maintenance zone, this is not considered in this chapter. The main focus is on station areas which, normally, have strict speed restrictions such that, often, no further speed limitations are applied. As a consequence, differences in departure or arrival time and cancelations are the only origin of planned delays.

Finally, because the size of the deviations from the original plan is bounded from above by the algorithm, see Section 9.5.1, only small planned delays occur. Therefore, it is assumed that these small timetable changes do not persuade passengers to take another train if theirs is not canceled. Moreover, due to the small timetable deviations, no rolling stock or crew schedule conflicts are expected to arise.

In the next section, an overview of the related literature about scheduling railway maintenance is given. Section 9.3 discusses the impact of the maintenance on the applied robustness function. Next, the applied extensions to the developed methodology to deal with the track possessions are discussed in Section 9.4. Finally, in Section 9.5 the computational results are presented and Section 9.6 concludes this chapter.

9.2 Literature about maintenance planning within a timetable

Since the maintenance schedule is assumed to be given, the literature study about maintenance is restricted to some references about scheduling railway infrastructure maintenance within a given train timetable. A good starting point for this study is the dissertation of Budai-Balke (2009). In her dissertation, Budai-Balke provides an extensive literature review as well as several exact and non-exact solution methods to minimize the maintenance cost. Next to minimizing the maintenance cost, which is also used in Budai et al. (2006), minimizing the planned delays that are needed to obtain a feasible timetable (Higgins 1998; Higgins et al. 1999; Lake et al. 2002) and a combination of the two (Peng et al. 2011) are used to schedule the maintenance activities within a given train timetable. Another approach is discussed in Van Zante-De Fokkert et al. (2007). To improve the safety, cyclic preventive maintenance with a period of 4 weeks is scheduled during the moments of low or no operation in the Dutch timetable. Although this is a very interesting approach, we do not schedule maintenance (only) at night times since we consider large maintenance (or construction) tasks which last (much) longer such that interference with trains in operation is inevitable.

The interference between the trains and the infrastructure maintenance is twofold. On the one hand, there is the foreseen hinder due to the overlap in blocking times that leads to planned delays, and on the other hand, there is the unforeseen hinder since the reduced capacity increases the system's sensitivity to delay propagation. While there are some papers that focus on the former, as far as we are aware, our approach is the first that explicitly studies the increased sensitivity to delay propagation when updating the train schedule for a set of already planned track possessions. This is confirmed by the quotation of the beginning of this chapter and the observation that in Lake et al. (2002), the focus is on maintenance planning and not on the train timetabling.

Forsgren et al. (2013) start from an original timetable and use a mixed integer problem (MIP) to schedule the track possessions in the (modifiable) train schedule. The track possessions they consider last for some hours and have flexible start and end times. The objective used in the MIP is threefold: (i) minimize the number of maintenance conflicts in the timetable, (ii) minimize the number of canceled trains, and (iii) minimize the planned delays. No robustness-related aspects are considered. Moreover, in Section 9.5.2, we show that the objective of minimizing the planned delays can have a negative impact on the passenger service. Next to varying the start times of the maintenance actions, Forsgren et al. allow four measures to reach their goals: moving a train in time, rerouting, canceling, and slowing down trains, for example, to wait until the maintenance is over (if realistic).

As input for their MIP model, all candidate alternative routes and timings for each train need to be listed. Like mentioned in their paper as well as in Caimi et al. (2005) for a comparable problem, this approach only works for limited instances because otherwise the computation times explode. Since we want to consider complex and busy networks with many routing possibilities for each of the 80 hourly trains, we consider their approach inadequate for our study.

Next to the approach and the objectives, there is a difference in the type of networks that is studied. Brucker et al. (2002) study one line with heterogeneous traffic, Louwerse et al. (2014) use two merging lines with several stations, and in Forsgren et al. (2013), a network that contains both single and double track sections is analyzed. In this chapter, the NSC with its multiple parallel lines and large switch zones is used as case study. Nevertheless, the usage of the algorithm is not restricted to station areas and more general networks can be handled.

For a network consisting of long-haul single tracks and a system with freight trains without rigid timetables, the train scheduling and maintenance planning is done simultaneously in Albrecht et al. (2013) and Pudney et al. (2004). Track possessions are modeled as pseudo trains and then the train timetable, including the pseudo trains, is computed with the objective to minimize the planned delays. The planned delays originate from the trains that are planned to wait at the entrance of a closed single track and from the extra setups and time losses that are caused by interruptions of the track possessions to let the waiting trains pass.

In case of a large disruption or urgent, unforeseen maintenance, real-time interventions are required. Delay management strategies that concern the transfers such as in Dollevoet et al. (2012) and Meng et al. (2011), or train reallocation actions to satisfy the passenger demand (Canca et al. 2012) as well as real-time rerouting and rescheduling as studied by, among others, Corman et al. (2010), Louwerse et al. (2014), and Tamura et al. (2013) become appropriate. Another important aspect that must be considered then is the rolling stock (re-)scheduling like is done in, for example, Cadarso et al. (2013) and Nielsen et al. (2012). Often, the rolling stock scheduling is being revised in cases of (planned) maintenance (see Budai-Balke 2009; Cacchiani et al. 2008a, for some examples), however, this is beyond the scope of this dissertation.

9.3 Planned maintenance and the definition of robustness

In Chapter 3, our definition of robustness is presented. A railway system is said to be robust if it minimizes the real weighted travel time (RWTT) of the passengers. To measure the RWTT, the durations of all events along the journey of all passengers are weighted and summed. In case of maintenance actions that interfere with the regular train services, the timetable gets updated and planned delays arise. Since planned delays extend the normal journey time by an early departure and/or a postponed arrival, the adaptation to the updated timetable is considered as an extra action of a journey. As a consequence, this adaptation is added to Table 9.1 which then extends Table 3.1 such that all events that influence the real travel time in case of maintenance interventions are included in this table.

Table 9.1: Overview of the different passenger actions in case of planned maintenance, their events, nature, and assigned weights. Since planned delays extend the regular journey time by an early departure and/or a postponed arrival, the adaptation to the updated timetable is included in this table. The entries in the column *nature* indicate whether the occurrence and/or duration of the corresponding event is Deterministic or Stochastic.

action	**event**	**nature**	**weight**
board	board	D	0
	cancelation	S	3
dwell	minimum necessary ride and dwell time	D	1
	useful ride and dwell supplements	S	1
	non-useful ride and dwell supplements	S	2
transfer	minimum necessary ride and transfer time	D	1
	useful RTS and transfer buffer	S	1
	non-useful RTS and transfer buffer	S	2
	missed transfer	S	3
alight	minimum necessary ride time	D	1
	useful RTS	S	1
	non-useful RTS	S	2
	arrival delays	S	3
adaptation to maintenance	**planned (cancelation) delays**	**D**	**1**

Analogue to the derivation of the robustness measures in Chapter 3, the formulas (3.1)-(3.5) to compute the RWTT with its variants and the robustness scores Rob_1 and Rob_2 can be updated to cope with the planned delays. Doing so, planned (cancelation) delays are included in the real travel time and thus also accounted for in the robustness of the system. Since these delays cause an inconvenience to the passengers, similar to, but to a smaller extent than real delays, planned delays naturally are part of the RWTT and the robustness. Therefore, the planned delays are included in the robustness values of the results from the computational experiments in Section 9.5.

9.4 Methodology

In order to avoid maintenance conflicts, the algorithm from the previous chapters needs to be adapted. The routing module from Chapter 5 should become capable of handling train-track possession conflicts and the platforming module of Chapter 7 can be used to reroute trains around the closed track sections[39]. To distinguish the original version of the modules from the maintenance version, the term *maintenance* will be added to the name of the modules. In case a certain train cannot be (re-)routed away from the closed track, a cancelation action follows. This happens in the *cancelation module*. Since canceling trains is no part of the robustness' optimization, this feature is new to the algorithm.

Shifting trains in time or swapping the order of two trains does not directly help to solve a maintenance conflict. Therefore, the timetabling module is only used as internal timetabling module within the *maintenance platforming module*. Note, however, that after the first objective of avoiding all maintenance conflicts is met, the original algorithm from the previous chapters is used to improve the robustness of the obtained schedule and then the timetabling module is used as before. Obviously, the maintenance-free property should not be violated then. Each change to the timetable gives a cost for the passengers in terms of planned delays. Since the total delays and the robustness of a solution are only measured after the algorithm has finished, no trade-off between a timetable change that improves the spreading to avoid conflicts and the resulting planned delays is made in the algorithm. Therefore, time window constraints that restrict the maximal size of the timetable deviations are imposed for each train. As a result, the size of the planned delays remains limited and the assumption that passengers do not change trains because of the timetable update remains acceptable.

[39]Since the focus of the platforming module shifts from selecting a new platform with a new route to the selection of a new maintenance-free route, the updated version of the platforming module is used to *reroute* trains.

In the course of the algorithm, maintenance conflicts are solved step by step. Unlike train-train conflicts, intermediate solutions with train-track possession conflicts are allowed. However, the final solution should be maintenance-free as well as conflict-free. In order to minimize the number of cancelations as much as possible, the algorithm iterates between the *maintenance routing module* and the maintenance platforming module. The cancelation module is called only when no routing or rerouting action succeeded in decreasing the number of trains that are scheduled over the closed tracks. In this case, one train is taken out of operation and then the algorithm returns to the maintenance routing module to check whether the freed capacity can be used to (re-)route another train. The outline of the adapted methodology is given in Algorithm 9.1.

Throughout this section, we will refer to the example network that is depicted in Figure 9.1. This network consists of two parallel tracks that are connected by switches. In total, three trains run over this network: two from left to right following route r_1 (trains t_1 and t_2) and one in the other direction on the lower track: train t_3 that follows route r_2. According to the original timetable, trains t_1 and t_3 enter the network simultaneously. There is one track possession that is scheduled on the upper track between the two switches.

Algorithm 9.1 Framework of the procedure to avoid maintenance conflicts

input: timetable and track possessions
while there are still maintenance conflicts **do**
  solve the maintenance version of the train routing problem (TRP)
  apply the maintenance platforming module
  if no maintenance conflict is avoided **then**
    start the cancelation module

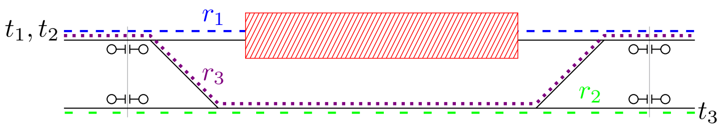


Figure 9.1: Example of a track possession that requires rescheduling. In this network with three trains running over it, a single track possession (shaded area) is scheduled on the upper track. The trains on the upper track, t_1 and t_2, follow route r_1. An alternative route (r_3) to avoid the track possession is indicated with the dotted line. The train on the lower track, train t_3, enters the system simultaneously with train t_1 and runs along route r_2.

9.4.1 The maintenance routing module

When solving the train routing problem (TRP) with the Kaufman-TRP model from Chapter 5, no changes in train timings or in the usage of inbound or outbound lines of the network are allowed. In case of a track possession, some parts of the infrastructure become unavailable. In many cases, adding the constraint that no train is allowed to run over the closed track section would make the solution infeasible. Consider the example of Figure 9.1. Given that train t_1 enters the network at the same time as train t_3, it is impossible to find a feasible routing solution in which none of the trains uses the closed track section. To avoid situations in which no solution is returned, this constraint is not enforced, but instead, an extra (penalty) cost term is added to the objective function such that using the closed track section is possible but incurs a high cost. The remaining train-track possession conflicts will be solved in the following iterations of the algorithm such that, in the end, a feasible solution arises. Nevertheless, intermediate solutions with maintenance conflicts are allowed.

The addition of the extra penalty term to the original objective function (5.23) goes as follows. Using the same notation as before, we define $P_{(t,r)}$ as the penalty of trainroute (t, r) which is 0 if route r does not pass a closed track section. The new objective function then becomes

$$\text{Minimize} \sum_{t \in T} \sum_{r \in R_t} z_{(t,r)} + P_{(t,r)} \cdot x_{(t,r)}. \tag{9.2}$$

Note that $x_{(t,r)}$ is the binary variable that is 1 if trainroute (t, r) is selected and 0 otherwise. The size of the penalties, when nonzero, is chosen according to a measure of importance. This measure is based on the type of the train. Ranked from important to less important, the following train types are used: international or high speed trains (type 4), intercity trains (type 3), interregional trains (type 2), and local trains (type 1). Potential peak hour trains are categorized according to their stopping regime. If trainroute (t, r) of train t with type $type_t$ conflicts with a track possession, its penalty equals $P_{(t,r)} = 1000 \cdot type_t$. This way, the penalty of an international train is (much) larger than that of a local train what creates an incentive to reroute the more important trains before the less important ones. Since our main concern is to avoid conflicts with the track possessions, the penalties are chosen such that they are much larger than the total spreading cost (5.23). As comment, we want to add that we are well aware of multiple other criteria to determine the penalties, for example, using passenger flow information. Although the one above is selected, adjusting this to another rule is straightforward.

No other changes are made to the mathematical model to solve the TRP. This means that when ignoring the maintenance, the routing solution is feasible at all times.

To make the routing module more efficient, some preprocessing is added in Chapter 5. If trainroute (t, r) has an incompatible trainroute that has no smaller minimum time spans, (t, r) is dominated and can be removed from the system without affecting the optimal objective function value. For example, ignoring the track possession in Figure 9.1, route r_3 for train t_1 is dominated by its original route r_1 since the minimum time span of trainroute (t_3, r_2) with (t_1, r_1) is larger than the minimum time span with (t_1, r_3), $B_{(t_1,r_1),(t_3,r_2)} = 15 > 0 = B_{(t_1,r_3),(t_3,r_2)}$, and analogous for train t_2. This means that during the preprocessing phase, trainroutes (t_1, r_3) and (t_2, r_3) are pruned. In case of the track possession, however, this is unwanted since then the alternative route becomes unavailable. Although it is worse with respect to the spreading, (t_1, r_3) does not incur the penalty cost while its dominator (trainroute (t_1, r_1)) does incur the penalty. To avoid the pruning of (t_1, r_3), a rule is added that forbids removing a dominated trainroute that does not run over the track possession while its dominator does. As a consequence, (t_1, r_3) and (t_2, r_3) are not pruned. A similar rule can be applied to the route dominance technique of Section 5.1.1.

Applying the maintenance routing module to the example above, train t_2 gets rerouted along r_3 and thus avoids the track possession. For train t_1, this is not possible since otherwise a train-train conflict with train t_3 would arise. In the next steps of the algorithm, a solution for train t_1 is searched.

9.4.2 The maintenance platforming module

As said before, the maintenance platforming module corresponds to the platforming module. The main differences are the shift of the focus towards rerouting instead of a platform change and the fact that the new route may not use any closed track section. After rerouting, the internal timetabling module is applied to restore possible train-train conflicts. For example, a small time shift of train t_1 can make the rerouting of train t_1 via r_3 conflict-free.

The shift of the focus towards rerouting results in a different construction procedure for the restricted candidate list (RCL). Trains that are scheduled to run over a closed track form the RCL in the maintenance platforming module (step (i) in Algorithm 9.2). In step (ii), the RCL is ordered based on the associated penalty costs (from large to small). Doing so, the most important trains, those with the highest penalties, are at the beginning of the list and are considered first. In the next step, a new route that does not incur the penalty

Algorithm 9.2 The maintenance platforming module

input: timetable, routing solution, and the track possessions
(i) add all trains that conflict with the track possessions to the RCL
(ii) sort the RCL in descending order of the size of the penalties
  for all $t \in$ RCL and **for all** new routes $r' \in R_t$ **do**
    if r' does not conflict with the track possessions **then**
      (iii) perform the internal timetabling module
      (iv$_\mathrm{a}$) **if** the solution is not train-train conflict-free **then**
        restore original solution
      (iv$_\mathrm{b}$) **else** return to the maintenance routing module with the new solution
(v) (if no feasible rerouting is found) go to the cancelation module

cost is selected for the first train in the list. Whether this new route uses a different platform or leads to more intersecting routes is not considered. After that, the internal timetabling module is applied to restore the feasibility. This is step (iii). If it did not succeed, the rerouting action is rejected, the original solution restored, and the next route for that train or, in case all candidate routes for the current train are considered, the next candidate train is selected (step (iv$_\mathrm{a}$)). If the solution turned out to be feasible (with respect to train-train conflicts), the rerouting action is accepted in step (iv$_\mathrm{b}$). This means that the usage of the closed track is avoided for one train. Similar to the platforming module, only a single rerouting action is applied before the algorithm returns to the maintenance routing module. If no rerouting was successful, the cancelation module is called (step (v)). A complete overview of the maintenance platforming module is given in Algorithm 9.2.

The internal timetabling phase that is used in step (iii) does not only solve train-train conflicts but also tries to improve the spreading cost. Nevertheless, the only criteria to accept a rerouting action is the avoidance of a maintenance conflict. As a consequence, the spreading cost (without the maintenance penalties) tends to grow.

9.4.3 The Cancelation module

Consider again the example of Figure 9.1. If for any reason, it is not possible to (re-)route train t_1 along route r_3 without avoiding a conflict with train t_3, the cancelation module can help to remedy this: cancel train t_1 such that the resulting solution becomes feasible.

From the maintenance platforming module, the RCL, the ordered set of trains that conflict with a track possession, is known. In the cancelation module, the last element of the RCL, the train of lowest importance (lowest type), is taken out of operation. After that, the algorithm returns to the maintenance routing module. Similar to the criteria that determine the penalties, other cancelation orders can be selected, for example, in agreement with the railway operator or depending on the set of alternative trains. However, this is left aside here. Another aspect that is not considered in this chapter but that can happen in practice, is the fact that canceling a train does not mean that the train cannot ride at all. It suffices to cancel the trip(s) that are affected by the track possessions and to introduce a shuttle timetable in which trains run between their original endpoints and some intermediate station before the maintenance zone.

Note that we minimize the number of cancelations by only calling this module if the other modules become insufficient. Canceling trains creates (large) planned delays for the passengers since they (temporarily) have to take an alternative train or search for an alternative mode of transport. Therefore, by canceling one train at a time, we hope that the freed time slots allow to solve some of the remaining conflicts by the other modules at a smaller (passenger) cost.

From the moment a maintenance-free solution is found, be it after the maintenance routing module, the maintenance platforming module, or after the cancelation of a train, the original settings of the algorithm are restored and the algorithm continues. Thus from this point on, the timetabling module is activated again and the RCL construction in the platforming module goes as explained in Chapter 7. To prevent new train-track possession conflicts, the maintenance routing module is not replaced by the original routing module and no candidate new route that is considered in the platforming module may use the closed tracks. To avoid large differences between the original timetable and the updated version, the upper bound on the maximal size of the timetable changes that are applied in the internal timetabling module remain valid. In the following section, the influence of these bounds on the performance of the algorithm and the final result is discussed.

9.5 The impact of maintenance in the North-South connection (NSC)

The compact and highly used network connecting the main stations of Brussels is chosen as case study. As original timetable, the final timetable from Chapter 7 is used. The three specific cases of track maintenance that are studied are indicated in Figure 9.2. In case 1, one of the six platform tracks in the Central station is closed for maintenance. In the original solution, 15 on a total number of 80 trains, are scheduled to use this platform. The platforms in the Central station have a fixed orientation. Thus only two alternative platforms exist for all northbound trains. All trains, except two international trains, dwell at the central station for one minute. Since there is a headway of 2 minutes between departure and arrival in the Central station, a theoretical maximum of 41 trains per hour can be scheduled over these two platforms. There are, however, 42 northbound trains in the original timetable what means that at least one train needs to be canceled. We come back to this issue in Section 9.5.3. The second case consists of the closing of two parallel tracks that connect the upper platforms of the North station with two platforms of the Central station. This affects 22 trains. Note that the switches at both ends of the closed tracks remain open such that all platforms are still usable. This is in contrast with the third case in which the schedule of (at least) 35 trains needs to be modified in order to cope with the track possessions. In this case, part of the grid near station Midi is closed for maintenance. As a consequence, four incoming or outgoing lines (here: platforms in station Midi) become non-reachable. As alternative for these platforms, the neighboring incoming or outgoing lines at each side are selected.

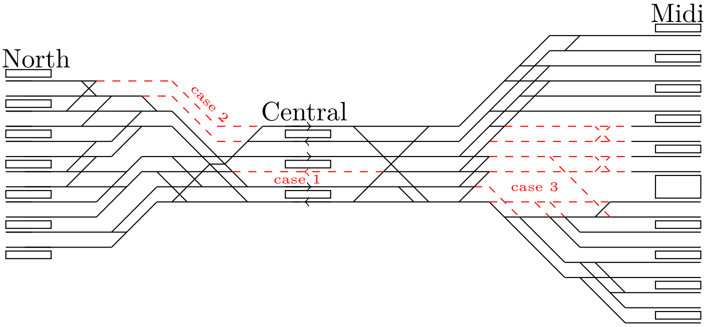


Figure 9.2: The different track possessions in the NSC case study. The three maintenance zones are indicated with dashed lines. In case 1, one of the platforms in the bottleneck station is closed. Case 2 consists of two parallel track possessions. For the third case, a whole group of tracks is taken out of usage which makes some incoming lines unavailable.

9.5.1 Algorithm results

As indicated before, time window restrictions are used to limit the deviations from the original schedule. Let e^{orig} (e^{up}) be the event time of event e in the original (updated) timetable, expressed in minutes. Then timetable changes up to, for example, 4 minutes correspond to a time window restriction of the form

$$e^{\text{up}} \in [e^{\text{orig}} - 4, e^{\text{orig}} + 4]. \tag{9.3}$$

To simplify the notation, $[-4, 4]$ is used instead of (9.3). In total, three different types of time windows are considered: (i) only postponing events, for example time windows of the form $[0, 4]$, (ii) advancing and postponing events such as in (9.3), and (iii) stepwise increasing the size of the time windows when no successful rerouting action is found. Initially, no shift is allowed and instead of a cancelation, the time window bounds are increased by one minute before the algorithm returns to the maintenance platforming module. Time windows of this type are denoted as $[0, \varepsilon]$ or $[-\varepsilon, \varepsilon]$, with ε the bound that gets increased. Since no differences worth mentioning are found for timetable deviations larger than 5 minutes, this is used as upper bound for ε.

In Figure 9.3, the decrease of the number of maintenance conflicts is shown per iteration for case 1 with the $[-4, 4]$ time window constraint for each train. In general, more than one conflict can be solved during the first iterations. For example, in iterations 1 and 3 where, respectively, 7 and 2 maintenance conflicts are solved. This comes from the maintenance routing module. The one by one reduction in the middle comes from the maintenance platforming module which can solve conflicts among trains and becomes more efficient than the maintenance routing module after some iterations. At the end, if the maintenance platforming module cannot solve the last conflicts anymore, the cancelation module is applied to take trains out of operation iteratively and in increasing order of importance.

After the second rerouting, the maintenance routing module is able to route an extra train around the track possessions. Also in the maintenance routing module of iteration 4 an extra train of type 2 is routed around the maintenance zone, but this goes at the cost of an extra type 3 train that uses the closed track instead. This shows that the rerouting of one train may create a potential for another train to be rerouted. The same holds for cancelations. For example, in time window scenarios $[0, 1]$ and $[0, 2]$, both for case 1, the cancelation of a train enabled the rerouting of a more important train. This proves that canceling trains one by one is useful to minimize the number of cancelations.

An overview of the global reduction in the number of conflicts for all three cases and with the $[-4, 4]$ time window constraint for each train is given in Figure 9.4.

The large improvement in the beginning (due to the maintenance routing module) and the stepwise reduction that follows afterwards can also be seen in this figure. Figures 9.3 and 9.4 show that the first objectives of the passenger service, obtaining a feasible schedule with a limited number of cancelations are achieved.

When varying the upper and lower bounds for the time windows, more or less freedom arises and the number of canceled trains changes. This can be seen in Figure 9.5 where the number of canceled trains for each time window scenario is shown. In general, more freedom helps to decrease the number of canceled trains. There are, however, some exceptions which are due to the algorithm getting stuck in a local optimum. For example, when events can be postponed with one minute in case 3 (time window scenario $[0, 1]$), only 3 trains need to be canceled, while scenario $[0, 2]$ results in 4 cancelations.

As discussed in Section 7.3.1, the best solution of all iterations is selected as final outcome. Although tight time window constraint restrict the solution space, improved objective function values (9.2) are achieved in the iterations after all maintenance conflicts are solved. On average, less iterations are made than for the regular NSC case study. In order to fully evaluate the impact of the maintenance actions on the performance, more results than the number of cancelations is needed. For this, the results of the simulation module are used.

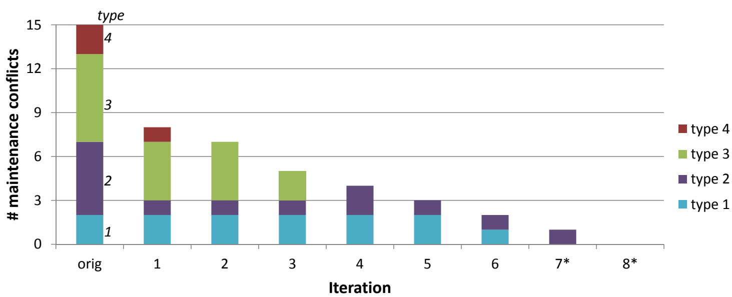


Figure 9.3: Evolution of the number of trains per type that conflict with the track possessions for maintenance case 1 with the $[-4, 4]$ time window constraint. The larger the type number, the more important the train and the higher the penalty. Iteration *orig* stands for the original timetable. Each iteration consists of a call to the maintenance routing module followed by a rerouting action or cancelation. When a train is canceled in a certain iteration, an asterisk is added to the label on the X-axis.

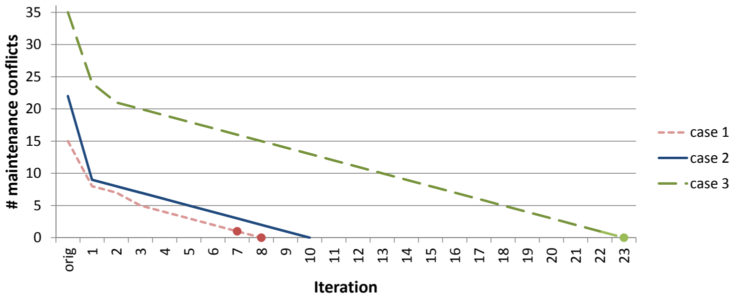


Figure 9.4: Evolution of the number of trains that conflict with the track possessions per case. During the computations, the [-4,4] time windows constraints are used. Each iteration consists of a call to the maintenance routing module followed by the maintenance platforming module (and the cancelation module). The bullets indicate that a train is canceled in this step.

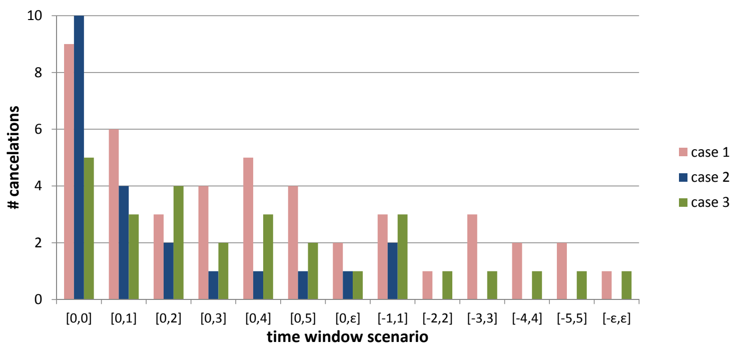


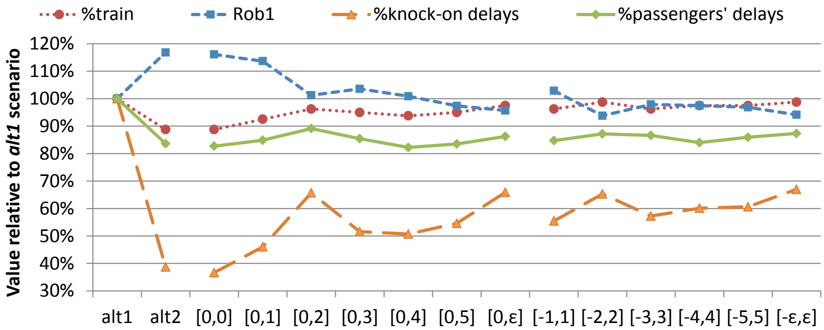
Figure 9.5: The number of canceled trains for each time window per case. Time window $[0, \varepsilon]$ ($[-\varepsilon, \varepsilon]$) corresponds to the scenario where the upper (and lower) bound is gradually increased (to maximum 5 minutes) instead of calling the cancelation module. This way, extra planned delays are only inserted in the updated timetable when this effectively leads to avoiding cancelations.

9.5.2 Simulation results

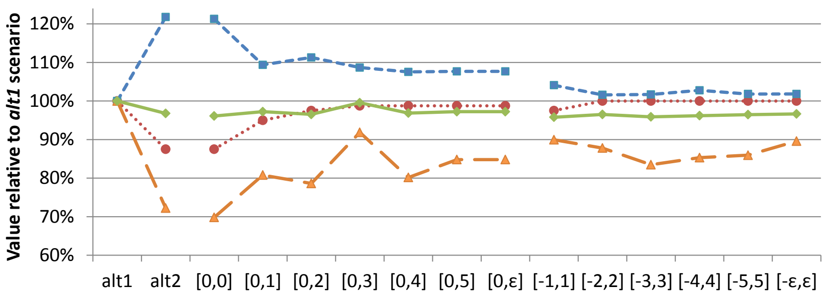
An analysis of the results is made using a graphical representation of the simulated performance when delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$ is used. Not only enables this an easier assessment of the different components that contribute to the final robustness score, it also allows a quick comparison of the different solutions. To evaluate the quality of our approach and the different scenarios given by the time windows, two alternative solutions (alt_1 and alt_2) are used as reference solutions in the evaluation of the performance of our algorithm. The first alternative solution equals the original timetable in which all trains are kept but the ones that conflict with the maintenance actions are rerouted along maintenance-free routes without timetable changes. Therefore, the Kaufman-TRP model is adapted to allow train-train conflicts at the cost of a penalty. This is done in a similar way as explained in Section 9.4.1 for train-track possession conflicts. As a consequence, the resulting alt_1 timetable can be infeasible. For the second alternative solution, alt_2, all trains that are scheduled to run over a closed track after the maintenance routing module is applied, are canceled simultaneously. The difference with the $[0, 0]$ scenario is twofold. When using time window $[0, 0]$, the maintenance platforming module attempts to solve train-track possession conflicts by considering alternative platforms (without modifying the timetable) and a rerouting action or cancelation is alternated with the maintenance routing module, while in the alt_2 scenario, all trains are canceled at once. In total, the alt_2 approach resulted in 9, 10 and 25 cancelations for cases 1-3, respectively.

The simulation results are visualized in Figures 9.6 and 9.7. Figure 9.6 contains information about the relative number of trains that are running in each of the updated schedules (*%train*) and thus represents the number of cancelations. Also the robustness score Rob_1 and the relative amount of *knock-on delays* and *passengers' delays* are plotted. All results are computed relative to the alt_1 scenario that serves as reference system. In part (a) of this figure, the correlation between the number of trains and the other performance indicators is clearly visible. As seen in Section 8.1.1, the throughput in the NSC benefits from having less trains. This translates into a reduced amount of knock-on delays and passengers' delays. However, since removing trains from the system brings along cancelation delays that influence the RWTT, this positive effect is undone for solutions with many canceled trains like alt_2 and the $[0, 0]$ scenario in cases 1 and 2. The allowance of timetable shifts not only reduces the cancelations that are necessary to solve the train-track possession conflicts, but also helps to decrease the amount of propagated delays and the total passengers' delays. Based on the Rob_1 robustness scores, the best solutions can be detected. In case 1, the robustness of the alt_1 solution, which is used as reference, is improved by the solutions of some time window scenarios from which time window scenario $[-2, 2]$ is the best with a reduction in RWTT of 6%. For the second case, no timetable returned by our algorithm improves the (infeasible) alt_1 solution. We ascribe this to the combination of the robustness of the original timetable, which is the *final* timetable of Chapter 7, and the limited impact of the track possessions on the available capacity. Since each train can still use its original combination of platforms, the closure of the tracks and corresponding rerouting action of the alt_1 scenario can be seen as a disturbance to a system that is optimized with respect to robustness. As a consequence, the overall impact remains limited. Just as in case 1, the unavailability of some platforms makes the reduction of the available capacity in case 3 is larger than in case 2. Nevertheless, the substantial set of alternative routes around the maintenance zone reduces the need for canceling trains. In case 3, all solutions that are outputted by our algorithm, independent of the time window constraints, improve the alternative scenarios. The largest improvement in robustness is achieved with time window constraint $[0, 1]$ and amounts 12.3% compared with the alt_1 scenario. Although solutions with only one canceled train exist for case 3, canceling three trains results in a lower RWTT. The same holds for case 1. In such a situation, the final decision about which timetable will be selected lies the hands of the railway companies.

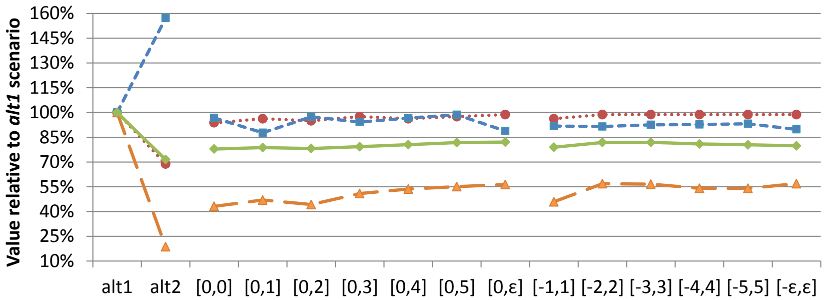
Figure 9.7 provides more insight in the ratios of the various types of delays that influence the robustness. For this, all values are scaled such that the total cumulative amount of delays in scenario alt_1 equals 100%. The real or unplanned delays per passenger (*passengers' delays*) are plotted together with the *planned delays* and the planned *cancelation delays*. Notice that the robustness scores in Figure 9.6 show an equal pattern as the cumulative delays, but that Rob_1 is computed based on the RWTT, while no weights are used in Figure 9.7. The reason is that the latter figure's subject is the size of the delays and not the passengers' valuation of travel time. For all cases, there is a reduction in passengers' delays compared with the (infeasible) alt_1 scenario, but together with the planned delays and cancelation delays, the cumulative amount of delays hardly gets below the reference amount. Nevertheless, by recognizing the annoyance of the unplanned passengers' delays and thus accounting for the passengers' perception, the robustness of some of the newly computed timetables is considered better than that of the original timetable. In this figure, the impact of more freedom for rescheduling on all types of delays becomes clear. For example, the cancelation delays decrease at the cost of some planned delays. Together with Figure 9.6, this shows that minimizing the planned delays, as is often done in literature, is not always better for the cumulative amount of delays and for the passengers' valuation of travel time. Also after a maintenance-free solution is found and the algorithm continues its improvement process, planned delays can arise. Figure 9.8 is used to gain some insight in this.



(a) Case 1

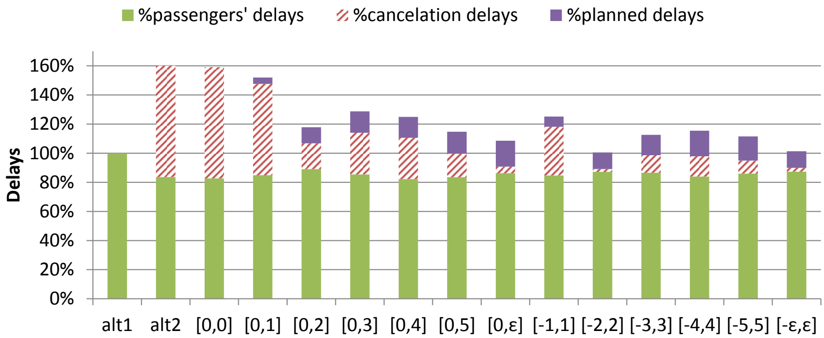


(b) Case 2

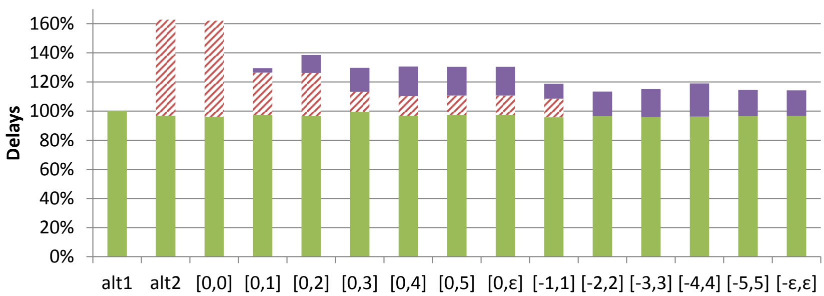


(c) Case 3

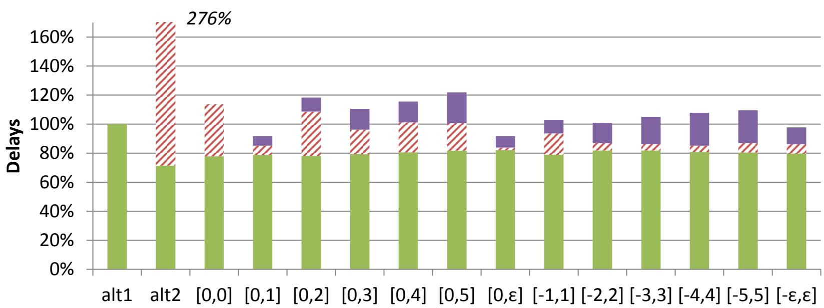
Figure 9.6: Output of the simulation module for the three maintenance cases. All results are obtained using delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. For the two alternative solutions and the various time window scenarios, the relative number of *trains*, the Rob_1 robustness score, and the relative amount of *knock-on delays* and *passengers' delays* are included in each graph.



(a) Case 1



(b) Case 2



(c) Case 3

Figure 9.7: Output of the simulation module for the three maintenance cases. All results are obtained using delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. The total cumulative delays the passengers experience are divided in the three types of delays: unplanned *passengers' delays*, *cancelation delays*, and *planned delays*.

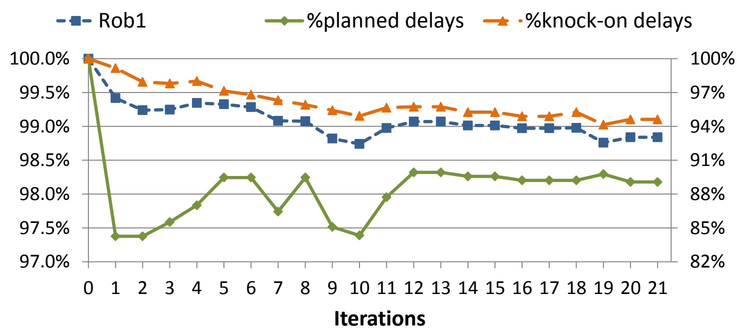


Figure 9.8: Impact of planned delays on the robustness and the propagation of delays for each iteration after all maintenance conflicts are solved for case 1 with the $[-2, 2]$ time window constraint.

In Figure 9.8, the *planned delays* (right Y-axis), the amount of *knock-on delays* (also right Y-axis), and the robustness (Rob_1, left Y-axis) are plotted. The X-axis starts from the moment the first maintenance-free solution is found (iteration 0) and ranges till the last iteration before the search stabilizes. This plot is obtained for case 1 with the time window constraint that gives the best results for this case: time window $[-2, 2]$. Along the improvement process, there is a clear downward trend for the robustness and the amount of propagated delays. The planned delays fluctuate more along the iterations. The best solution, with respect to the robustness value, is reached after the tenth iteration. This is the solution that is used for this scenario in the previous figures. The difference with the outcome of iteration 19 is less than 0.02% in Rob_1 value. There are, however, much more planned delays in the latter solution. This shows once more that deviations from the original timetable, which imply more scheduled inconvenience for the passengers are not necessarily bad for the robustness of the updated timetable. Another conclusion that can be drawn based on Figure 9.8 is that it is worthwhile to continue the robustness' improvement process after all maintenance conflicts are solved. In only 7 of the 39 scenarios[40] that are computed, the first maintenance-free solution (iteration 0) turned out to be the best one. For all others, the extra optimization improved the solution.

At last, we want to dig a little deeper in the propagation of real delays. Table 9.2 allows to see at which location in the network most of the conflicts occur and how the impact of the maintenance reduces in the course of the algorithm. For each case, the alt_1 scenario is compared with the best solution for this case. Since the number of operated trains influences the results, this is added to

[40] For each of the three cases, 13 different time windows are considered. This gives 39 scenarios in total.

Table 9.2: The amount of propagated delays (in minutes) in the stations and on the grids for the three maintenance cases in delay scenario $\left(E_{|T|/2}, 0, P^{(0.5)}_{|T|/2}, 0\right)$. The results in the first row correspond to the original timetable without maintenance (see Table 7.6). For each case, the numbers for the alt_1 scenario are presented together with the results for the best solution of the algorithm for this case. The subscript *feas* indicates the first feasible timetable and *opt* is used to indicate the results of the best found solution.

		#trains	North	grid NC	Central	grid CM	Midi
original timetable		80	4.7	1.6	9.7	1.6	0.9
case 1	alt_1	80	5.4	6.5	54.2	8.4	1.0
	$[-2,2]_{\text{feas}}$	79	5.5	5.5	34.9	5.0	1.0
	$[-2,2]_{\text{opt}}$	79	4.8	5.0	33.8	4.5	1.2
case 2	alt_1	80	5.8	10.5	19.7	3.2	1.0
	$[-2,2]_{\text{feas}}$	80	6.0	8.2	17.6	3.1	1.2
	$[-2,2]_{\text{opt}}$	80	6.6	8.1	16.9	2.8	1.0
case 3	alt_1	80	4.6	1.6	15.5	28.4	30.6
	$[0,1]_{\text{feas}}$	77	5.1	1.3	13.4	13.9	8.0
	$[0,1]_{\text{opt}}$	77	5.3	0.9	12.7	12.1	7.0

the table. Comparing the different cases, one sees the increase in delays at the location of the track possessions for the corresponding case. Also in the neighboring zones, a spill back effect can be noticed. The extremely high amount of propagated delays in the Central station for case 1 is remarkable. This is not surprising since it is this bottleneck that is affected by the maintenance. The reduction in knock-on delays for the best time window scenarios shows the strength of our algorithm.

9.5.3 Capacity usage in the Central station

As can be seen in Figure 9.5, only one cancelation is required to avoid all maintenance conflicts for time windows $[-2, 2]$ and $[-\varepsilon, \varepsilon]$ in case 1. As discussed above, this means that the maximum number of northbound trains that can be scheduled in a conflict-free way through the NSC is effectively scheduled. Surprisingly, time window scenario $[-2, 2]$ gives the best solution for case 1. The results in Table 9.2 provide some insight. It shows that no real explosion of the amount of knock-on delays arises at that station. This is a consequence of some simplifications that are made when modeling the NSC and of a few assumptions of the simulation module. Nevertheless, since all simulations for the NSC case study are performed with the same model and the results are always considered relatively to one another, the final outcome can be used.

Consider the total passengers' delays for case 1 in Figure 9.7. Scenario alt_1, which runs in overcapacity, gives considerably more delays than the other scenarios. The two solutions that use the full capacity, time windows $[-2, 2]$ and $[-\varepsilon, \varepsilon]$, lead to worse results with respect to passengers' delays than the other ones, but the differences are limited. Since the cancelation delays for the removed trains in time window scenarios $[-2, 2]$ and $[-\varepsilon, \varepsilon]$ are quite low, these solutions prove to be the best with respect to the robustness.

This discussion clarifies the main drawback of the presented approach to avoid planned track possessions. Since one of the objectives of passenger service is to minimize the number of cancelations and, in the algorithm, no direct assessment is made of a cancelation on the system's sensitivity to knock-on delays or its robustness, which is another objective of passenger service, some mismatch between these objectives may arise. As indicated before, based on the set of canceled trains, railway companies can decide which solution they prefer. Other options to intervene in the decision process are by deciding on how the impact of a canceled train on the RWTT should be modeled, or by using different weights in the cancelation module in order to influence the cancelation order.

9.6 Conclusion

In this chapter, the approach that is developed in the previous chapters is used as basis for an algorithm that can reschedule the system to enable planned maintenance actions. Doing so, the applicability of the robustness definition and the presented methodology to other timetable-related problems, which is questioned in the second part of research question 4 is illustrated.

Starting from an original timetable and a set of track possessions, the objective of this chapter is to minimize the worsening in service level to the passengers. This corresponds to a minimal number of canceled trains and an updated timetable that is as robust as possible while a feasible schedule is required. Therefore, the set of actions that part the RWTT is extended with an action that is typical for this problem: adapting one's travel behavior to the updated schedule. This fits perfectly in the definition of robustness as presented in Chapter 3.

The algorithm to avoid the maintenance conflicts is based on that of the previous chapters and extended with the cancelation module that is only performed if no other module was successful.

In the course of the algorithm, train-track possession conflicts are solved step by step, and a cancelation action is only applied if no acceptable rerouting is found. This helps to minimize the number of canceled trains. Moreover, by canceling only one train at a time, the possibility to reroute a more important train is enabled in some situations.

The computational results show that allowing timetable changes is useful in the quest for a maintenance-free and robust solution. Although shifts in the arrival and departure times of a train induce planned delays to the passengers, it is shown in Section 9.5.2 that extra planned delays do not necessarily increase the RWTT. This also proves the advantages of a robustness oriented approach above rescheduling and rerouting without considering the robustness. By continuing the algorithm after all maintenance conflicts are solved, a better solution is found for a large majority of the considered scenarios.

The discussion of the solution with full usage of the available capacity in the Central station pointed at a drawback of this approach. Because the impact of cancelations is not considered in the algorithm, solutions that are suboptimal from the viewpoint of the railway company may arise. However, since the algorithm is not specifically fine-tuned for the maintenance avoidance, this does not alter the fact that it is suitable for problems like the one considered in this chapter.

Chapter 10

Conclusions and further perspectives

Trains can have delays. Yes, but they can also have less delays.

this dissertation

10.1 Conclusions

Trains can have delays. These are the first words of the introduction of this dissertation. Somewhat later follows: *Yes, but they can also have less delays. At least, according to the conclusions of this dissertation.*

At the time where sustainable mobility is a hot topic, efficiency is high up on the agendas and passenger service is more important than ever. This makes robustness one of the key elements to avoid staying behind with an unreliable railway system and low punctuality numbers. Therefore, the Belgian railway infrastructure manager Infrabel raised the question of developing principles and techniques that are applicable on the tactical level and make the Belgian railway timetable more robust.

Inspired by the observation that bottlenecks, and more specifically large and complex station areas, are the main sources of delays, the main objective of this dissertation is to improve the robustness of a railway system in large and complex station areas. To reach this goal, some research questions are formulated. The first one is inspired by the lack of a formal definition of railway robustness.

> **Research Question 1:** How to define the robustness of a railway system? What is the contribution of different elements in obtaining a robust railway system?

During a review of the state-of-the-art literature, multiple interpretations of robustness are encountered. Where some authors see the propagation of delays as main aspect of robustness, some others consider the transfer reliability more important. Often, one says that the robustness can be increased by adding slack to the timetable. Although all interpretations are value adding, several drawbacks are identified. Therefore, we tried to extract the essence of robustness and used this as basis to formulate an all-embracing definition of robustness. Simply said, robustness is the ability of a system to keep its level of service towards the clients of that system. In passenger railways, the clients are the passengers. As a consequence, it is crucial to account for the passengers and their valuation of travel time when optimizing the robustness. This allowed us to formulate our all-embracing and practically usable definition of railway robustness.

> ***A railway system that is robust against the daily occurring, small disturbances minimizes the real weighted travel time (RWTT) of the passengers.***

Studying the robustness of a railway timetable at the tactical level makes sense if only frequent but small disturbances are considered. The definition states that a system is robust if and only if the average real duration of all passengers' trips is perceived as small as possible. It is not the published travel time that matters, but the actual time it takes to reach your destination. Thus, the faster the passengers are transported, even in case of small disturbances, the more robust the system.

To measure the robustness of a certain railway system, all events that can influence the required time to complete a trip are seen as components of the RWTT. Each component has its own duration and gets a weight that represents the valuation of that particular kind of travel time. As a result, the robustness of railway systems is easily measured through simulation. Moreover, open questions such as how much timetable slack is needed to optimize the robustness become superfluous since the robustness function automatically assesses the impact of planned travel time extensions on the real travel time. Another advantage is that, although they are sometimes seen as opposing, robustness and efficiency go hand in hand with this approach. As a consequence, railway operators, railway infrastructure managers, and passengers benefit from this new and comprehensive definition.

To illustrate the practical applicability of the presented definition of robustness, an analysis of the robustness of the 2010 timetable for Belgium is made based on actual delay data. Although this example was useful, the emphasis of this dissertation lies at densely used railway bottlenecks such as the Brussels' North-South connection (NSC), the link between the stations Brussels North and Brussels Midi in Belgium. Due to its central location in the Belgian railway network and the large number of trains that run through it, a good local performance works beneficial for the robustness of the entire railway system. Therefore, the second and third research question focus on the NSC.

> **Research Question 2:** How to deal with the limited capacity in the North-South connection (NSC) in Brussels? How to make the timetable for the NSC more robust?

To answer this research question, an algorithm is developed that improves the robustness in railway bottlenecks. Based on an objective function that spreads the usage of resources in time, optimization of train routing through the network is addressed, an integrated approach to exploit the potential of changing train orders and schedules is built, and the potential of modifying the platform allocations can be evaluated.

The objective of our methodology is to increase the minimum time span between any two trains in a station area. Although this objective function does not aim directly at robustness, it is a commonly used approach to indirectly improve the performance of a railway system. Its effectiveness is shown by a comparison with another objective from the literature and by an analysis of the key performance indicators' evolution during the algorithm. This showed that the drawback of replacing our robustness functions by the spreading objective remains limited to some variation in the performance indicators.

The developed algorithm consists of three parts. (i) First, there is the routing module in which the train routing problem (TRP) is solved such that each train gets exactly one route assigned. In order to facilitate the calculations, two preprocessing phases are worked out. The first considers the set of blocked links per route and prunes routes that block more links than necessary. The second dominance rule detects trainroutes that can be replaced without worsening the objective function value. In the end, the TRP is modeled as a mixed integer linear problem (MILP) and solved to optimality for the remaining set of candidate trainroutes.

(ii) The second part of the algorithm, the timetabling module, always follows upon the routing module. Since similar studies have shown that exact techniques only work for small instances, which the NSC surely is not, heuristics are used.

Based on a tabu search framework, an integrated approach that exploits the potential of changing the schedule or the train sequences is implemented.

(iii) The third and last part of the developed algorithm is the platforming module. When solving the TRP or modifying the timetable gives no improvement anymore, a platform change is applied. The number of intersecting routes and the differences in platform occupations are used to select promising platform change candidates. Analyzing the output of the algorithm, we observed the need for an extra (internal) timetable optimization before evaluating a platform change. The same holds for an order swap in the timetabling module. With this extra optimization, possible conflicts between trains are solved and the potential of the move can be explored before the move itself is evaluated. Thanks to this feature, considerably better solutions are found.

The impact of each step of the algorithm on the robustness is assessed. This assessment showed that improving the robustness of the train routing is highly valuable in large and complex station areas like the NSC. Although the overall impact of only changing the routes is limited, a remarkable reduction in propagation of delays on the grids is realized. When the timetable and platform allocations are also modified by the algorithm, the performance improves in all aspects and in all parts of the considered network. For example, the amount of knock-on delays is more or less halved, there is a reduction of about 7.2% in passengers' delays, and the RWTT shortens with more than 3%. Although this reduction may seem modest, the accompanying performance indicators show its value. Moreover, when only considering the stochastic components of the RWTT, a reduction of about 7.5% is measured. Next to the fact that less conflicts occur, fewer trains are harmed, and the travel times for the passengers goes down. Also the standard deviations of the various performance indicators decrease such that the system becomes more stable. As a consequence, we conclude that the robustness of the overall system is considerably improved. Thus, we may claim that the developed algorithm is able to fully exploit the potential of an integrated optimization of routing decisions, train sequences and schedules, and platform allocations.

It is a commonly accepted idea that the limited capacity of Brussels Central station is the main cause of the high number of conflicts and the large amount of secondary delays that arise in the NSC. The high number of passenger movements and the fact that the blocking times in the station are larger than for a regular section result in a large threat on head-tail conflicts. This is also visible by looking at the amount of delay propagation on the grids or in the stations. However, the results of this dissertation show that the Central station is not the only cause of delays. When comparing the impact of a single disturbance in one of the grids and that of a single dwell delay at the Central station, the impact of the former is larger for the original timetable. Due to the crisscross of lines that merge, intersect, and split in the station area of Brussels, many trains share resources, for sure in the grids. Thanks to the impact of the algorithm, in the final solution, there is a reduced interaction between the routes through the grids and an improved spreading of the blocking times. As a consequence, the hindrance caused by a grid conflict has decreased considerably. The more balanced platform occupations and better spread arrival times also reduced the impact of station conflicts, but to a smaller extent. In the end, for the final result of the algorithm, a dwell delay in the Central station caused more knock-on delays than a grid conflict, but the harm for the passengers remained larger for the latter. From this, it is concluded that not only the limited capacity of the Central station badly influences the performance, but also the line planning and routing solution are responsible. Nevertheless, once the timetabling, routing, and platforming are improved, the Central station can be considered as the main bottleneck of the Brussels' area. This conclusion is also supported by the findings of the third research question.

> **Research Question 3:** What is the effect on the robustness of a number of structural measures for the North-South connection (NSC)?

To answer this question, a "*what if*"-study is made to evaluate the influence of measures that alter the available capacity or the capacity usage in the NSC. Therefore, some strategic decisions about the network and the line planning are made to relax the throughput at the Central station. Next to the direct impact of each measure, the developed algorithm is applied to improve the robustness of each newly created system.

The best results are found by removing some trains from the system. Since this measure lowers the capacity usage in the entire NSC and not only in the Central station like the other measures do, this is not surprising. Thanks to the lower occupation rate of the system, the performance of the trains improves considerably. For the passengers, however, the improvements are only valid on the condition that they have a comparable alternative. The measures that altered the stopping pattern for some trains or extended the infrastructure in the Central station also improve the normal situation. The impact, however, is more local and mainly limited to relieving the Central station. When the passage through the Central station is kept unchanged, but the flow towards the bottleneck is grouped into corridors such that the interaction between trains on the grids reduces, the robustness also improves. Together with the findings of the system with less trains, this supports the conclusion about the importance of the routing through the grids as part of the bottleneck.

By performing these experiments, it is also shown that the developed methodology can cope with changing resources or different settings. A further illustration of the applicability of the methodology and validation of its usefulness is the subject of the last research question.

> **Research Question 4:** Can the developed approach be used for other bottlenecks in the network or for other timetable-related problems?

The validation question is answered by considering two extra case studies. The network of the first case study is based on an extension of the NSC such that it includes the beginning of the open tracks, the outer grids, and the entrances to the shunt yards. Doing so, the full interaction between the trains in the entire station area of Brussels is captured. Next to that, the modeling of the extended network is more detailed and corresponds better to reality. For the second case study, the station area of Antwerp is subjected to the developed algorithm. This case study is characterized by a high capacity usage by a heterogeneous fleet of trains and by the fact that all but four platforms in the Central station are dead-ending.

For both case studies, the simulation output predicts an improvement in robustness of nearly 10%. This means that the RWTT shortens with about one tenth thanks to the changes made by the algorithm. One of the causes of this large improvement is the reduction in the amount of propagated delays of 35% for the entire Brussels' station area and 42% for the case study of Antwerp. Together with the fact that both cases are completely different, this leads to the conclusion that the developed algorithm is generally applicable to improve the robustness in any densely used, large and complex railway station.

The obtained results are confirmed by a validation study using one of the Belgian railway infrastructure manager's commercial simulation packages. To this end, the study area of Antwerp is extended such that it contains an extra stopping station for each train. Next to the fact that this did not give real problems with respect to the feasibility of the improved timetable, our conclusions are validated since the improvements to the system that they obtained are of the same order of magnitude as the results we present.

The general applicability of the developed approach to other timetable-related problems is illustrated by considering the problem of rescheduling in case of planned unavailability of some infrastructure. Planned closing some parts of the infrastructure reduces the capacity of a railway system and makes it more vulnerable to conflicts and delay propagation. Starting from an original timetable and a set of track possessions, the objectives that are addressed are to minimize the number of canceled trains and to keep the updated timetable as robust as possible while a feasible planning is required. The originally developed approach is used as basis for an algorithm that meets these objectives. Also in this algorithm, special attention is given to the potential of a change for future improvements.

In the new settings, it is assumed that the original timetable is in operation. As a consequence, any change to the original schedule impacts the passengers who need to adapt their travel behavior. In the light of passenger service, a trade-off is made between this inconvenience and the delays that occur in practice when operating according to the new schedule. By updating the set of actions that compose the RWTT, it is shown that the presented definition of robustness adapts naturally to the new settings. A comparison of the first feasible solution with the final outcome of the algorithm illustrated that the addition of extra timetable changes to improve the spreading helps to decrease the RWTT in case of some closed resources. This proves the advantage of a robustness oriented approach above rescheduling and rerouting without considering robustness as in literature.

Studying the case of planned track closures illustrates the applicability of the new definition of robustness and the developed approach to other timetable-related problems.

Final conclusions

In order to improve the robustness of a railway system in large and complex station areas, a new and all-embracing definition of the robustness of a railway system is defined. Therefore, the contribution of different elements in obtaining a robust railway system are discussed. Based on this definition, we developed an approach to make the timetable for Belgium's main bottleneck more robust. By an integrated optimization of routing decisions, train sequences and schedules, and platform allocations, the railway system becomes more effective in managing the limited capacity in the North-South connection (NSC) in Brussels. In order to validate the presented approach and illustrate its general applicability, various cases that are characterized by different settings are studied. In light of this, the effect of a number of structural measures for the NSC on the robustness is assessed, other bottlenecks of the Belgian network are considered, and another timetable-related problem is studied.

Trains can have delays. Yes, but they can also have less delays.

10.2 Further perspectives

During the research that preceded this dissertation, many questions were answered, but also many new questions were raised. Not all of these could be handled in this dissertation. In this section, we discuss some of these questions and formulate some ideas for further research.

Instead of directly reducing the real weighted travel time (RWTT), the optimization process aims at a better spreading of the trains. Although we have shown that minimizing the spreading cost is a rather good approach to improve the robustness, there is some room for improvement. Since the spreading cost does not account for the passengers while robustness is defined from a passengers' viewpoint, incorporating passenger-based weights in the spreading objective function can result in a better guiding of the algorithm. Another aspect that is not considered in the spreading objective is the expected delay of each train. The performance can benefit by increasing the minimum time span between two trains with fluctuating real arrival times compensated by a shorter headway buffer after a rather punctual train.

Due to the lack of adequate transfer data for the Brussels' area, no actions to improve the transfer reliability are taken in the algorithm. Similarly, no transfers are considered in the simulation module. In the current planning process and the used reference system, also no transfers are taken into account. On the one hand, restricting the timetabling module by implying a list of transfers will result in a higher spreading cost. On the other hand, improving the transfer reliability may lead to a considerably shorter RWTT. The risk, however, is that the share of the avoidance of missed transfers in the total robustness' improvement can be so large that the focus on the flow through the bottleneck may fade.

The internal timetabling module is used to exploit the potential of timetable changes or platform reassignments. Except in case of rerouting without platform changes, no potential checks are made upon considering the routes of the trains. As a consequence, the impact of allowing suboptimal routing solutions remains unclear. An interesting topic for future work is to incorporate some feedback loop from the timetabling module or platforming module in the routing module.

The analysis of the results of the different case studies is a source of inspiration for more thoughts for further research. For example, the impact of the changed platform allocations in measure 4 (Chapter 8) raises some questions: What can be the impact of allowing the platforming module to assign multiple trains at once to a new platform? Would the algorithm evolve towards grouping platforms of the same orientation? Are there other actions that can be taken to stimulate the concept of corridors?

The necessary turn around movements are one of the main causes of knock-on delays in the station area of Antwerp. In the developed algorithm, no special action is included to improve this. By extending the available turn around time, trains can get rid of delays and the new journey can start punctually. There is also a high benefit in changing the turn around combinations. If the rolling stock requirements allow it, allocating trains to a new line can help to obtain larger and more equally divided turn around times.

In the chapter about the maintenance actions, some of the infeasible timetables performed rather good. This can be inspiring for a study on the North-South connection (NSC). Similar to the real-time platform selection in the station of Schiphol in the Netherlands (Schaafsma et al. 2007) or to the more theoretical concept of flexible timetables (Caimi et al. 2011b; D'Ariano et al. 2008b), leaving some decisions to the operational level may increase the performance within the Brussels' area.

Bibliography

Adams, W. and T. Johnson (1994). “Improved linear programming based lower bounds for the quadratic assignment problem”. In: *Series in discrete mathematics and theoretical computer science*. Ed. by P. Pardalos and H. Wolkowic. Vol. 16. Quadratic assignment and related problems. American Mathematical Society, pp. 43–75.

Albrecht, A., D. Panton, and D. Lee (2013). “Rescheduling rail networks with maintenance disruptions using problem space search”. In: *Computers & Operations Research* 40, pp. 703–712.

Andersson, E., A. Peterson, and J. Törnquist Krasemann (2013). “Quantifying railway timetable robustness in critical points”. In: *Journal of Rail Transport Planning & Management* 3, pp. 95–110.

Barber, F., M. Abril, M. Salido, and L. Ingolotti (2007). “Survey of automated systems for railways management”. In: *Technical report, Universidad Politécnica de Valencia, Spain.*

Bates, J., J. Polak, P. Jones, and A. Cook (2001). “The valuation of reliability for personal travel”. In: *Transportation Research Part E: Logistics and Transportation Review* 37, pp. 191–229.

Bengtsson, L., R. Galia, T. Gustafsson, C. Hjorring, and N. Kohl (2007). “Railway crew pairing optimization”. In: *Algorithmic methods for railway optimization*. Ed. by F. Geraets, L. Kroon, A. Schöbel, D. Wagner, and D. Zaroliagis. Vol. 4359. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 126–144.

Ben-Tal, A. and A. Nemirovski (1998). “Robust convex optimization”. In: *Mathematics of Operations Research* 23, pp. 769–805.

Bertsimas, D. and M. Sim (2004). “The price of robustness”. In: *Operations Research* 52, pp. 35–53.

Bessas, A., S. Kontogiannis, and C. Zaroliagis (2009). “Incentive-compatible robust line planning”. In: *Robust and online large-scale optimization*. Ed. by R. Ahuja, H. Möhring, and C. Zaroliagis. Vol. 5868. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 85–118.

Biederbick, C. and L. Suhl (2007). "Decision support tools for customer-oriented dispatching". In: *Algorithmic methods for railway optimization*. Ed. by F. Geraets, L. Kroon, A. Schöbel, D. Wagner, and D. Zaroliagis. Vol. 4359. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 171–183.

Billionnet, A. (2003). "Using integer programming to solve the train-platforming problem". In: *Transportation Science* 37, pp. 213–222.

Brucker, P., S. Heitmann, and S. Knust (2002). "Scheduling railway traffic at a construction site". In: *OR Spectrum* 24, pp. 19–30.

Budai, G., D. Huisman, and R. Dekker (2006). "Scheduling preventive railway maintenance activities". In: *Journal of the Operational Research Society* 57, pp. 1035–1044.

Budai-Balke, G. (2009). "Operations research models for scheduling railway infrastructure maintenance". PhD thesis. Erasmus Universiteit Rotterdam, the Netherlands.

Büker, T. and B. Seybold (2012). "Stochastic modelling of delay propagation in large networks". In: *Journal of Rail Transport Planning & Management* 2, pp. 34–50.

Burkolter, D. (2005). "Capacity of railways in station areas using petri nets". PhD thesis. Swiss Federal Institute of Technology, Zurich, Switzerland.

Bussieck, M. (1998). "Optimal lines in public rail transport". PhD thesis. Technischen Universität Braunschweig, Germany.

Cacchiani, V., A. Caprara, and M. Fischetti (2012a). "A lagrangian heuristic for robustness, with an application to train timetabling". In: *Transportation Science* 46, pp. 124–133.

Cacchiani, V., A. Caprara, L. Galli, L. Kroon, G. Maróti, and P. Toth (2008a). "Recoverable robustness for railway rolling stock planning". In: *Proceedings of the 8th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS)*. Ed. by M. Fischetti and P. Widmayer. Vol. 9. OpenAccess Series in Informatics (OASIcs). Dagstuhl, Germany, pp. 113–125.

Cacchiani, V., A. Caprara, and P. Toth (2008b). "A column generation approach to train timetabling on a corridor". In: *4OR* 6, pp. 125–142.

Cacchiani, V., D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar (2014). "An overview of recovery models and algorithms for real-time railway rescheduling". In: *Transportation Research Part B: Methodological* 63, pp. 15–37.

Cacchiani, V. and P. Toth (2012b). "Nominal and robust train timetabling problems". In: *European Journal of Operational Research* 219, pp. 727–737.

Cadarso, L. and A. Marín (2011). "Robust rolling stock in rapid transit networks". In: *Computers & Operations Research* 38, pp. 1131–1142.

— (2012). "Integration of timetable planning and rolling stock in rapid transit networks". In: *Annals of Operations Research* 199, pp. 113–135.

Cadarso, L., A. Marín, and G. Maróti (2013). “Recovery of disruptions in rapid transit networks”. In: *Transportation Research Part E: Logistics and Transportation Review* 53, pp. 15–33.

Caimi, G., D. Burkolter, and T. Herrmann (2005). “Finding delay-tolerant train routings through stations”. In: *Operations Research Proceedings*. Ed. by H. Fleuren, D. Hertog, and P. Kort. Vol. 2004. Springer-Verlag, pp. 136–143.

Caimi, G., D. Burkolter, T. Herrmann, F. Chudak, and M. Laumanns (2009a). “Design of a railway scheduling model for dense services”. In: *Networks and spatial Economics* 9, pp. 25–46.

Caimi, G., F. Chudak, M. Fuchsberger, M. Laumanns, and R. Zenklusen (2011a). “A new resource-constrained multicommodity flow model for conflict-free train routing and scheduling”. In: *Transportation Science* 45, pp. 212–227.

Caimi, G., M. Fuchsberger, D. Burkolter, T. Herrmann, R. Wüst, and S. Roos (2009b). “Conflict-free train scheduling in a compensation zone exploiting the speed profile”. In: *Proceedings of 3rd International Conference on Railway Operations Modelling and Analysis (RailZurich)*. Zurich, Switserland.

Caimi, G., M. Fuchsberger, M. Laumanns, and M. Luethi (2012). “A model predictive control approach for discrete-time rescheduling in complex central railway station areas”. In: *Computers & Operations Research* 39, pp. 2578–2593.

Caimi, G., M. Fuchsberger, M. Laumanns, and K. Schüpbach (2011b). “Periodic railway timetabling with event flexibility”. In: *Networks* 57, pp. 3–18.

Canca, D., E. Barrena, A. Zarzo, F. Ortega, and E. Algaba (2012). “Optimal train reallocation strategies under service disruptions”. In: *Procedia - Social and Behavioral Sciences* 54, pp. 402–413.

Caprara, A., M. Fischetti, and P. Toth (2002). “Modeling and solving the train timetabling problem”. In: *Operations Research* 50, pp. 851–861.

Caprara, A., L. Galli, and P. Toth (2011). “Solution of the train platforming problem”. In: *Transportation Science* 45, pp. 246–257.

Caprara, A., L. Kroon, M. Monaci, M. Peeters, and P. Toth (2007). “Passenger railway optimization”. In: *Handbooks in operations research and management science*. Ed. by C. Barnhart and G. Laporte. Vol. 14. Elsevier, pp. 129–187.

Carey, M. and S. Carville (2000). “Testing schedule performance and reliability for train stations”. In: *Journal of the Operational Research Society* 51, pp. 666–682.

Carey, M. and I. Crawford (2007). “Scheduling trains on a network of busy complex stations”. In: *Transportation Research Part B: Methodological* 41, pp. 159–178.

Chakroborty, P. and D. Vikram (2008). “Optimum assignment of trains to platforms under partial schedule compliance”. In: *Transportation Research Part B: Methodological* 42, pp. 169–184.

Cicerone, S., G. D'Angelo, G. Di Stefano, D. Frigioni, A. Navarra, M. Schachtebeck, and A. Schöbel (2009). "Recoverable robustness in shunting and timetabling". In: *Robust and online large-scale optimization.* Ed. by R. Ahuja, H. Möhring, and C. Zaroliagis. Vol. 5868. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 28–60.

Cordeau, J.-F., P. Toth, and D. Vigo (1998). "A survey of optimization models for train routing and scheduling". In: *Transportation Science* 32, pp. 380–404.

Corman, F. (2010). "Real-time railway traffic management: Dispatching in complex, large and busy railway networks". PhD thesis. Technische Universiteit, Delft, the Netherlands.

Corman, F., A. D'Ariano, and I. Hansen (2014). "Evaluating disturbance robustness of railway schedules". In: *Journal of Intelligent Transportation Systems* 18, pp. 106–120.

Corman, F., A. D'Ariano, D. Pacciarelli, and M. Pranzo (2010). "A tabu search algorithm for rerouting trains during rail operations". In: *Transportation Research Part B: Methodological* 44, pp. 175–192.

Corman, F., A. D'Ariano, M. Pranzo, and I. Hansen (2011). "Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area". In: *Transportation Planning and Technology* 34, pp. 341–362.

Corman, F., R. Goverde, and A. D'Ariano (2009). "Rescheduling dense train traffic over complex station interlocking areas". In: *Robust and online large-scale optimization.* Ed. by R. Ahuja, R. Möhring, and C. Zaroliagis. Vol. 5868. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 369–386.

Cornelsen, S. and G. Stefano (2007). "Platform assignment". In: *Algorithmic methods for railway optimization.* Ed. by F. Geraets, L. Kroon, A. Schöbel, D. Wagner, and C. Zaroliagis. Vol. 4359. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 233–245.

D'Ariano, A. (2008). "Improving real-time train dispatching: Models, algorithms and applications". PhD thesis. Technische Universiteit, Delft, the Netherlands.

D'Ariano, A., F. Corman, D. Pacciarelli, and M. Pranzo (2008a). "Reordering and local rerouting strategies to manage train traffic in real time". In: *Transportation Science* 42, pp. 405–419.

D'Ariano, A., D. Pacciarelli, and M. Pranzo (2007). "A branch and bound algorithm for scheduling trains in a railway network". In: *European Journal of Operations Research* 183, pp. 643–657.

— (2008b). "Assessment of flexible timetables in real-time traffic management of a railway bottleneck". In: *Transportation Research Part C: Emerging Technologies* 16, pp. 232–245.

Delorme, X., X. Gandibleux, and J. Rodriguez (2009). “Stability evaluation of a railway timetable at station level”. In: *European Journal of Operational Research* 195, pp. 780–790.

Delorme, X., J. Rodriguez, and X. Gandibleux (2001). “Heuristics for railway infrastructure saturation”. In: *Electronic Notes in Theoretical Computer Science* 50, pp. 39–53.

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen (2011). “Defining robustness of a railway timetable”. In: *Proceedings of 4th International Conference on Railway Operations Modelling and Analysis (RailRome).* Rome, Italy.

— (2013). “Robust railway station planning: An interaction between routing, timetabling and platforming”. In: *Journal of Rail Transport Planning & Management* 3, pp. 68–77.

— (2014). “Improving the robustness in railway station areas”. In: *European Journal of Operational Research* 235, pp. 276–286.

Dollevoet, T. and D. Huisman (2014). “Fast heuristics for delay management with passenger rerouting”. In: *Public Transport* 6, pp. 67–84.

Dollevoet, T., D. Huisman, M. Schmidt, and A. Schöbel (2012). “Delay management with rerouting of passengers”. In: *Transportation Science* 46, pp. 74–89.

Engelhardt-Funke, O. and M. Kolonko (2004). “Analysing stability and investments in railway networks using advanced evolutionary algorithms”. In: *International Transactions in Operational Research* 11, pp. 381–394.

Fischetti, M. and M. Monaci (2009a). “Light robustness”. In: *Robust and online large-scale optimization.* Ed. by R. Ahuja, R. Möhring, and C. Zaroliagis. Vol. 5868. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 61–84.

Fischetti, M., D. Salvagnin, and A. Zanette (2009b). “Fast approaches to improve the robustness of a railway timetable”. In: *Transportation Science* 43, pp. 321–335.

Folkmann, M., J. Jespersen, and M. Nielsen (2007). “Estimates on rolling stock and crew in DSB S-tog based on timetables”. In: *Algorithmic methods for railway optimization.* Ed. by F. Geraets, L. Kroon, A. Schöbel, D. Wagner, and D. Zaroliagis. Vol. 4359. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 91–107.

Forsgren, M., M. Aronsson, and S. Gestrelius (2013). “Maintaining tracks and traffic flow at the same time”. In: *Journal of Rail Transport Planning & Management* 3, pp. 111–123.

Franke, B., B. Seybold, T. Büker, T. Graffagnino, and H. Labermeier (2013). “OnTime: Network-wide analysis of timetable stability”. In: *Proceedings of 5th International Conference on Railway Operations Modelling and Analysis*

(RailCopenhagen). Ed. by I. Hansen, O. Nielsen, A. Landex, J. Törnquist Krasemann, and J. Pachl. Copenhagen, Denmark.

Gendreau, M. (2003). "An introduction to tabu search". In: *Handbook of metaheuristics*. Ed. by F. Glover, G. Kochenberger, and F. Hillier. Vol. 57. International Series in Operations Research & Management Science. Springer New York, pp. 37–54.

Ghoseiri, K., F. Szidarovszky, and M. Asgharpour (2004). "A multi-objective train scheduling model and solution". In: *Transportation Research Part B: Methodological* 38, pp. 927–952.

Glover, F. and M. Laguna (1998). *Tabu search*. Kluwer Academic Publishers.

Goerigk, M., M. Schachtebeck, and A. Schöbel (2012). "Dependencies between line planning, timetabling and delay management: Simulations with the LinTim toolbox". In: *Technical report, Georg-August Universität, Göttingen, Germany.*

Goerigk, M. and A. Schöbel (2010). "An empirical analysis of robustness concepts for timetabling". In: *Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS)*. Ed. by T. Erlebach and M. Lübbecke. Vol. 14. OpenAccess Series in Informatics (OASIcs). Dagstuhl, Germany, pp. 100–113.

Goldratt, E. (1984). *The Goal: A process of ongoing improvement*. North River Press.

Goossens, J.-W., S. van Hoesel, and L. Kroon (2006). "On solving multi-type railway line planning problems". In: *European Journal of Operational Research* 168, pp. 403–424.

Goverde, R. (1998). "Synchronization control of scheduled train services to minimize passenger waiting times". In: *Proceedings of the 4th TRAIL Annual Congress, part 2*. Delft, the Netherlands.

— (2002). "Performance evaluation of network timtetables using PETER". In: ed. by J. Allan, E. Andersson, C. Brebbia, R. Hill, G. Sciutto, and S. Sone. Computers in Railways VIII. WIT Press, Southampton, pp. 1–10.

— (2005). "Punctuality of railway operations and timetable stability analysis". PhD thesis. Technische Universiteit, Delft, the Netherlands.

— (2007). "Railway timetable stability analysis using max-plus system theory". In: *Transportation Research Part B: Methodological* 41, pp. 179–201.

Goverde, R., I. Hansen, G. Hooghiemstra, and H. Lopuhaä (2001). "Delay distributions in railway stations". In: *Proceedings of the 9th World Conference on Transport Research (WCTR)*. Seoul, Korea.

Herrmann, T. (2006). "Stability of timetables and train routings through station regions". PhD thesis. Swiss Federal Institute of Technology, Zurich, Switzerland.

Higgins, A. (1998). "Scheduling of railway track maintenance activities and crews". In: *Journal of the Operational Research Society* 49, pp. 1026–1033.

Higgins, A., L. Ferreira, and M. Lake (1999). “Scheduling of railway track maintenance to minimise overall delays”. In: *Proceedings of the 14th International Symposium on Transportation and Traffic Theory*. Jerusalem, Israel.

Hooghiemstra, J., L. Kroon, M. Odijk, M. Salomon, and P. Zwaneveld (1999). “Decision support systems support the search for win-win solutions in railway network design”. In: *Interfaces* 29, pp. 15–32.

Huisman, D. (2007). “A column generation approach for the rail crew rescheduling problem”. In: *European Journal of Operational Research* 180, pp. 163–173.

Huisman, D., L. Kroon, R. Lentink, and M. Vromans (2005). “Operations research in passenger railway transportation”. In: *Statistica Neerlandica* 59, pp. 467–497.

Janecek, D. and F. Weymann (2010). “LUKS: Analysis of lines and junctions”. In: *Proceedings of the 12th World Conference on Transport Research (WCTR)*. Lisbon, Portugal.

Jensen, L. and A. Landex (2013). “Measuring robustness of timetables at stations using a probability distribution”. In: *Proceedings of 5th International Conference on Railway Operations Modelling and Analysis (RailCopenhagen)*. Ed. by I. Hansen, O. Nielsen, A. Landex, J. Törnquist Krasemann, and J. Pachl. Copenhagen, Denmark.

Kanai, S., K. Shiina, S. Harada, and N. Tomii (2011). “An optimal delay management algorithm from passengers’ viewpoints considering the whole railway network”. In: *Journal of Rail Transport Planning & Management* 1, pp. 25–37.

Kaspi, M. and T. Raviv (2013). “Service-oriented line planning and timetabling for passenger trains”. In: *Transportation Science* 47, pp. 295–311.

Kaufman, T. and F. Broeckx (1978). “An algorithm for the quadratic assignment problem using Bender’s decomposition”. In: *European Journal of Operational Research* 2, pp. 204–211.

Khosravi, B., J. Bennell, and C. Potts (2012). “Train scheduling and rescheduling in the UK with a modified shifting bottleneck procedure”. In: *Proceedings of the 12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS)*. Ed. by D. Delling and L. Liberti. Vol. 25. OpenAccess Series in Informatics (OASIcs). Dagstuhl, Germany, pp. 120–131.

Kroon, L., R. Dekker, and M. Vromans (2007a). “Cyclic railway timetabling: A stochastic optimization approach”. In: *Algorithmic methods for railway optimization*. Ed. by F. Geraets, L. Kroon, A. Schöbel, D. Wagner, and C. Zaroliagis. Vol. 4359. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 41–66.

Kroon, L., D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maróti, L. Schrijver, A. Steenbeek, and R. Ybema (2009). "The new Dutch timetable: The OR revolution". In: *Interfaces* 39, pp. 6–17.

Kroon, L., D. Huisman, and G. Maróti (2007b). "Railway timetabling from an operations research". In: *Econometric Institute Research Paper EI2007-22, Erasmus Universiteit Rotterdam, the Netherlands.*

— (2008a). "Optimisation models for railway timetabling". In: *Railway timetable & traffic: Analysis, modelling and simulation.* Ed. by I. Hansen and J. Pachl. Eurailpress, Hamburg, pp. 135–154.

Kroon, L., R. Lentink, and A. Schrijver (2008b). "Shunting of passenger train units: An integrated approach". In: *Transportation Science* 42, pp. 436–449.

Kroon, L. and G. Maróti (2008c). "Robust train routing". In: *Technical report, TR-0123, ARRIVAL.*

Kroon, L., G. Maróti, M. Retel Helmrich, M. Vromans, and R. Dekker (2008d). "Stochastic improvement of cyclic railway timetables". In: *Transportation Research Part B: Methodological* 42, pp. 553–570.

Kroon, L., H. Romeijn, and Zwaneveld P. (1997). "Routing trains through railway stations: Complexity issues". In: *European Journal of Operational Research* 98, pp. 485–498.

Kunimatsu, T., C. Hirai, and N. Tomii (2012). "Train timetable evaluation from the viewpoint of passengers by microsimulation of train operation and passenger flow". In: *Electrical Engineering in Japan* 181, pp. 51–62.

Lake, M. and L. Ferreira (2002). "Minimising the conflict between rail operations and infrastructure maintenance". In: *Proceedings of the 15th International Symposium on Transportation and Traffic Theory.* Ed. by M. Taylor. Transportation and Traffic Theory in the 21st Century. Oxford, UK, pp. 63–80.

Lawler, E. (1963). "The quadratic assignment problem". In: *Management Science* 9, pp. 586–599.

Lee, C.-K. (1998). "Delay analysis for train operation". In: ed. by B. Mellitt, R. Hill, J. Allan, G. Sciutto, and C. Brebbia. Computers in Railways VI. WIT Press, Southampton, pp. 513–522.

Liebchen, C. (2004). "Symmetry for periodic railway timetables". In: *Electronic Notes in Theoretical Computer Science* 92, pp. 34–51.

— (2008). "The first optimized railway timetable in practice". In: *Transportation Science* 42, pp. 420–435.

Liebchen, C., M. Lubbecke, R. Möhring, and S. Stiller (2009). "The concept of recoverable robustness, linear programming recovery, and railway applications". In: *Robust and online large-scale optimization.* Ed. by R. Ahuja, R. Möhring, and C. Zaroliagis. Vol. 5868. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 1–27.

Liebchen, C. and R. Möhring (2007). “The modeling power of the periodic event scheduling problem: Railway timetables - and beyond”. In: *Algorithmic methods for railway optimization*. Ed. by F. Geraets, L. Kroon, A. Schöbel, D. Wagner, and D. Zaroliagis. Vol. 4359. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 3–40.

Liebchen, C., M. Proksch, and F. Wagner (2008). “Performance of algorithms for periodic timetable optimization”. In: *Computer-aided systems in public transport*. Ed. by M. Hickman, P. Mirchandani, and S. Voss. Vol. 600. Lecture Notes in Economics and Mathematical Systems. Springer Berlin Heidelberg, pp. 151–180.

Liebchen, C., M. Schachtebeck, A. Schöbel, S. Stiller, and A. Prigge (2010). “Computing delay resistant railway timetables”. In: *Computers & Operations Research* 37, pp. 857–868.

Lindfeldt, A. (2013). “Heterogeneity measures and secondary delays on a simulated double-track”. In: *Proceedings of 5th International Conference on Railway Operations Modelling and Analysis (RailCopenhagen)*. Ed. by I. Hansen, O. Nielsen, A. Landex, J. Törnquist Krasemann, and J. Pachl. Copenhagen, Denmark.

Lindner, T. and T. Zimmermann (2005). “Cost optimal periodic train scheduling”. In: *Mathematical Methods of Operations Research* 62, pp. 281–295.

Liu, S. and E. Kozan (2009). “Scheduling trains as a blocking parallel-machine job shop scheduling problem”. In: *Computers & Operations Research* 36, pp. 2840–2852.

Louwerse, I. and D. Huisman (2014). “Adjusting a railway timetable in case of partial or complete blockades”. In: *European Journal of Operational Research* 235, pp. 583–593.

Lusby, R., J. Larsen, M. Ehrgott, and D. Ryan (2011a). “Railway track allocation: Models and methods”. In: *OR Spectrum* 33, pp. 843–883.

Lusby, R., J. Larsen, D. Ryan, and M. Ehrgott (2011b). “Routing trains through railway junctions: A new set-packing appraoch”. In: *Transportation Science* 45, pp. 228–245.

Mackie, P., S. Jara-Díaz, and A. Fowkes (2001). “The value of travel time savings in evaluation”. In: *Transportation Research Part E: Logistics and Transportation Review* 37, pp. 91–106.

Marín, A., J. Mesa, and F. Perea (2009). “Integrating robust railway network design and line planning under failures”. In: *Robust and online large-scale optimization*. Ed. by R. Ahuja, R. Möhring, and C. Zaroliagis. Vol. 5868. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 273–292.

Maróti, G. (2006). “Operations research models for railway rolling stock planning”. PhD thesis. Technische Universiteit, Eindhoven, the Netherlands.

Mascis, A. and D. Pacciarelli (2002). “Job shop scheduling with blocking and no-wait constraints”. In: *European Journal of Operational Research* 143, pp. 498–517.

Mazzarello, M. and E. Ottaviani (2007). “A traffic management system for real-time traffic optimisation in railways”. In: *Transportation Research Part B: Methodological* 41, pp. 246–274.

Meng, L. and X. Zhou (2011). “Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach”. In: *Transportation Research Part B: Methodological* 45, pp. 1080–1102.

Mesa, J., F. Perea, and G. Laporte (2013). “A reasonable approach for locating new stations in railway networks”. In: *Proceedings of the 8th Triennial Symposium on Transportation Analysis (TRISTAN).* San Pedro de Atacama, Chile.

Middelkoop, D. and M. Bouwman (2001). “Simone: Large scale train network simulations”. In: *Proceedings of the 33rd Conference on Winter Simulation (WSC).* Arlington, Virginia: IEEE Computer Society, pp. 1042–1047.

Middelkoop, D. and L. Loeve (2006). “Simulation of traffic management with FRISO”. In: ed. by J. Allan, C. Brebbia, A. Rumsey, G. Sciutto, and S. Sone. Computers in Railways X. WIT Press, Southampton, pp. 501–509.

Nachtigall, K. (1996). “Periodic network optimization with different arc frequencies”. In: *Discrete Applied Mathematics* 69, pp. 1–17.

Nachtigall, K. and S. Voget (1997). “Minimizing waiting times in integrated fixed interval timetables by upgrading railway tracks”. In: *European Journal of Operational Research* 103, pp. 610–627.

Nash, A. and D. Huerlimann (2004). “Railroad simulation using OpenTrack”. In: ed. by J. Allan, C. Brebbia, R. Hill, G. Sciutto, and S. Sone. Computers in Railways IX. WIT Press, Southampton, pp. 45–54.

Nielsen, L., L. Kroon, and G. Maróti (2012). “A rolling horizon approach for disruption management of railway rolling stock”. In: *European Journal of Operational Research* 220, pp. 496–509.

Pacciarelli, D. and M. Pranzo (2001). “A tabu search algortihm for the railway scheduling problem”. In: *4th Metaheuristics International Conference (MIC).* Porto, Portugal, pp. 159–163.

Pachl, J. (2008). “Timetable design principles”. In: *Railway timetable & traffic: Analysis, modelling and simulation.* Ed. by I. Hansen and J. Pachl. Eurailpress, Hamburg, pp. 9–42.

Peeters, J. and S. Stynen (2008). “De Noord-Zuid verbinding onder Brussel: Slagader of flessenhals? (The North-South connection under Brussels: Artery or bottleneck?) (in Dutch)”. In: *Mondig Mobiel* 83, pp. 14–17.

Peeters, L. (2003). “Cyclic railway timetable optimization”. PhD thesis. Erasmus Universiteit Rotterdam, the Netherlands.

Peng, F., S. Kang, X. Li, and Y. Ouyang (2011). “A heuristic approach to the railroad track maintenance scheduling problem”. In: *Computer-Aided Civil and Infrastructure Engineering* 26, pp. 129–145.

Pudney, P. and A. Wardrop (2004). “Generating train plans with problem space search”. In: *Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport (CASPT)*. San Diego, United States.

Rodriguez, J. (2007). “Enhancement of train routing and scheduling at a terminal station with constraints programming and temporal decomposition”. In: *Technical report, RR-07-731-FR, RECIFE.*

Salido, M., F. Barber, and L. Ingolotti (2012). “Robustness for a single railway line: Analytical and simulation methods”. In: *Expert Systems with Applications* 39, pp. 13305–13327.

Savelberg, F. and P. Bakker (2010). *Betrouwbaarheid en robuustheid op het spoor*. Kennisinstituut voor Mobiliteitsbeleid. Ministerie van Infrastructuur en Milieu. The Hague, the Netherlands.

Schaafsma, A. (2001). “Dynamisch railverkeersmanagement (Railway dynamic traffic management) (in Dutch)”. PhD thesis. Technische Universiteit, Delft, the Netherlands.

Schaafsma, A. and M. Bartholomeus (2007). “Dynamic traffic management in the Schiphol bottleneck”. In: *Proceedings of 2nd International Conference on Railway Operations Modelling and Analysis (RailHannover)*. Hannover, Germany.

Schittenhelm, B. (2011). “Creation of a framework for railway timetable optimization criteria”. In: *Proceedings of 4th International Conference on Railway Operations Modelling and Analysis (RailRome)*. Rome, Italy.

Schmidt, M. and A. Schöbel (2014). “Timetabling with passenger routing”. In: *OR Spectrum*. DOI: 10.1007/s00291-014-0360-0.

Schöbel, A. (2005). “Locating stops along bus or railway lines: A bicriteria problem”. In: *Annals of Operations Research* 136, pp. 211–227.

Schöbel, A. and A. Kratz (2009). “A bicriteria approach for robust timetabling”. In: *Robust and online large-scale optimization*. Ed. by R. Ahuja, R. Möhring, and C. Zaroliagis. Vol. 5868. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 119–144.

Schrijver, A. and A. Steenbeek (1994). “Dienstregelingontwikkeling voor Railned (Timetable construction for Railned) (in Dutch)”. In: *Technical report, C.W.I. Centrum Wiskunde & Informatica, Amsterdam, the Netherlands.*

Sels, P., T. Dewilde, D. Cattrysse, and P. Vansteenwegen (2011a). “Deriving all passenger flows in a railway network from ticket sales data”. In: *Proceedings of 4th International Conference on Railway Operations Modelling and Analysis (RailRome)*. Rome, Italy.

— (2013a). “A passenger knock-on delay model for timetable optimisation”. In: *Proceedings of 3rd International Conference on Models and Technologies for*

Intelligent Transportation Systems (MT-ITS). Ed. by T. Albrecht, B. Jaekel, and M. Lehnert. Dresden, Germany, pp. 377–387.

Sels, P., T. Dewilde, D. Cattrysse, and P. Vansteenwegen (2013b). "Expected passenger travel time for train schedule evaluation and optimization". In: *Proceedings of 5th International Conference on Railway Operations Modelling and Analysis (RailCopenhagen)*. Ed. by I. Hansen, O. Nielsen, A. Landex, J. Törnquist Krasemann, and J. Pachl. Copenhagen, Denmark.

Sels, P., T. Dewilde, P. Vansteenwegen, and D. Cattrysse (2011b). "Automated, passenger time optimal, robust timetabling, using integer programming". In: *Proceedings of 1st International Workshop on High-speed and Intercity Railways (IWHIR 2011)*. Ed. by Y.-Q. Ni and X.-W. Ye. Vol. 148. Lecture Notes in Electrical Engineering. Springer-Verlag Berlin Heidelberg, pp. 87–92.

Sels, P., P. Vansteenwegen, T. Dewilde, D. Cattrysse, B. Waquet, and A. Joubert (2014). "The train platforming problem: The infrastructure management company perspective". In: *Transportation Research Part B: Methodological* 61, pp. 55–72.

Sels, P., B. Waquet, T. Dewilde, D. Cattrysse, and P. Vansteenwegen (2011c). "Calculation of realistic railway station capacity by platforming feasibility checks". In: *Proceedings of 2nd International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. Ed. by F. Viti, C. Tampère, and B. Immers. Leuven, Belgium.

Serafini, P. and W. Ukovich (1989). "A mathematical model for periodic scheduling problems". In: *SIAM Journal on Discrete Mathematics* 2, pp. 550–581.

Shafia, M., M. Aghaee, S. Sadjadi, and A. Jamili (2012). "Robust train timetabling problem: Mathematical model and branch and bound algorithm". In: *IEEE Transactions on Intelligent Transportation Systems* 13, pp. 307–317.

Siefer, T. (2008). "Simulation". In: *Railway timetable & traffic: Analysis, modelling and simulation*. Ed. by I. Hansen and J. Pachl. Eurailpress, Hamburg, pp. 155–169.

Siefer, T. and A. Radtke (2005). "Railway simulation, key for better operation and optimal use of infrastucture". In: *Proceedings of 1st International Conference on Railway Operations Modelling and Analysis (RailDelft)*. Delft, the Netherlands.

Snelders, M., B. Immers, and I. Wilmink (2004). "De begrippen betrouwbaarheid en robuustheid nader verklaard (A detailed statement about the concepts of reliability and robustness) (in Dutch)". In: *Colloquium Vervoersplanologisch Speurwerk*. Zeist, the Netherlands.

Stynen, S. (2010). *Reizigerstellingen 2009 (Passenger counts 2009) (online, in Dutch)*. http://www.treintrambus.be/actueel/blog/1216-opstapcijfers.html.

Szpigel, B. (1973). "Optimal train scheduling on a single track railway". In: *Operation research*. Ed. by M. Ross. Vol. 72. Amsterdam, the Netherlands, pp. 343–351.

Takeuchi, Y., N. Tomii, and C. Hirai (2007). "Evaluation method of robustness for train schedules". In: *Quarterly Report of Railway Technical Research Institute* 48, pp. 197–201.

Tamura, K., N. Tomii, and K. Sato (2013). "An optimal rescheduling algorithm from passengers' viewpoint based on mixed integer programming formulation". In: *Proceedings of 5th International Conference on Railway Operations Modelling and Analysis (RailCopenhagen)*. Ed. by I. Hansen, O. Nielsen, A. Landex, J. Törnquist Krasemann, and J. Pachl. Copenhagen, Denmark.

Tomii, N., Y. Tashiro, N. Tanabe, C. Hirai, and K. Muraki (2005). "Train rescheduling algorithm which minimizes passengers' Dissatisfaction". In: *Innovations in applied artificial intelligence*. Ed. by M. Ali and F. Esposito. Vol. 3533. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 829–838.

Törnquist, J. and J. Persson (2005). "Train traffic deviation handling Using tabu search and simulated annealing". In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS)*. Hawaii, United States.

Tourwé, S. (2014). "Microscopische simulatie van een dienstregeling: conflictdetectie en evaluatie (Microscopic simulation of a timetable: conflict detection and evaluation) (in Dutch)". In: *Internship report, KU Leuven, Belgium.*

Tseng, Y.-Y., P. Rietveld, and E. Verhoef (2005). "A meta-analysis of valuation of travel time reliability". In: *Colloquium Vervoersplanologisch Speurwerk*. Antwerps, Belgium.

Van Oort, N. (2011). "Service reliability and urban public transport design". PhD thesis. Technische Universiteit, Delft, the Netherlands.

Van Zante-De Fokkert, J., D. Den Hertog, F. Van Den Berg, and J. Verhoeven (2007). "The Netherlands schedules track maintenance to improve track workers' safety". In: *Interfaces* 37, pp. 133–142.

Vansteenwegen, P. (2008). "Planning in tourism and public transportation: Attraction selection by means of a personalised electronic tourist guide and train transfer scheduling". PhD thesis. KU Leuven, Belgium.

Vansteenwegen, P. and D. Van Oudheusden (2006). "Developing railway timetables which guarantee a better service". In: *European Journal of Operational Research* 173, pp. 337–350.

— (2007). "Decreasing the passenger waiting time for an intercity rail network". In: *Transportation Research Part B: Methodological* 41, pp. 478–492.

Veelenturf, L., D. Potthoff, D. Huisman, and L. Kroon (2012). “Railway crew rescheduling with retiming”. In: *Transportation Research Part C: Emerging Technologies* 20, pp. 95–110.

VIA-CON, VIA Consulting & Development GmbH (2013). *LUKS: Analysis of lines and junctions.* http://www.via-con.de/development/luks/.

Vromans, M. (2005). “Reliability of railway systems”. PhD thesis. Erasmus Universiteit Rotterdam, the Netherlands.

Vromans, M., R. Dekker, and L. Kroon (2006). “Reliability and heterogeneity of railway services”. In: *European Journal of Operational Research* 172, pp. 647–665.

Wardman, M. (2004). “Public transport values of time”. In: *Transport Policy* 11, pp. 363–377.

Wong, R., T. Yuen, K. Fung, and J. Leung (2008). “Optimizing timetable synchronization for rail mass transit”. In: *Transportation Science* 42, pp. 57–69.

Yuan, J. (2006). “Stochastic modelling of train delays and delay propagation in stations”. PhD thesis. Technische Universiteit, Delft, the Netherlands.

Zhou, X. and M. Zhong (2007). “Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds”. In: *Transportation Research Part B: Methodological* 41, pp. 320–341.

Zwaneveld, P. (1997). “Railway planning: Routing of trains and allocation of passenger lines”. PhD thesis. Erasmus Universiteit Rotterdam, the Netherlands.

Zwaneveld, P., L. Kroon, and S. van Hoesel (2001). “Routing trains through a railway station based on a node packing model”. In: *European Journal of Operations Research* 128, pp. 14–33.

Zwaneveld, P., L. Kroon, H. Romeijn, M. Salomon, S. Dauzère-pérès, S. van Hoesel, and H. Ambergen (1996). “Routing trains through railway stations: Model formulation and algorithms”. In: *Transportation Science* 30, pp. 181–194.

Curriculum Vitae

Thijs Dewilde
Born on 25 April 1986 in Brugge, Belgium.

Education

2009 - present	PhD Engineering Sciences – KU Leuven
2004 - 2009	Master of Mathematics – KU Leuven – Magna cum laude Master's thesis about the *profit based latency problem* supervised by prof. dr. Frits Spieksma
2007 - 2008	Erasmus exchange – Aarhus University (Denmark)

List of publications

Articles in internationally reviewed academic journals

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2014) "Reducing the impact of infrastructure maintenance on the performance of railway systems". To be submitted soon.

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2014) "Improving the robustness in railway station areas". In: *European Journal of Operational Research* 235(1), pp. 276–286.

Sels, P., P. Vansteenwegen, **T. Dewilde**, D. Cattrysse, B. Waquet, and A. Joubert. (2014) "The train platforming problem: The infrastructure management company perspective". In: *Transportation Research Part B: Methodological* 61, pp. 55–72.

Sels, P., **T. Dewilde**, D. Cattrysse, and P. Vansteenwegen. (2014) "Reducing the passenger travel time in practice by the automated construction of a robust railway timetable". Submitted to *Transportation Research Part B: Methodological*.

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2013) "Robust railway station planning: An interaction between routing, timetabling and platforming". In: *Journal of Rail Transport Planning & Management* 3(3), pp. 68–77.

Dewilde, T., D. Cattrysse, S. Coene, F.C.R. Spieksma, and P. Vansteenwegen. (2013) "Heuristics for the traveling repairman problem with profits". In: *Computers & Operations Research* 40(7), pp. 1700–1707.

Article in academic book, internationally recognized scientific publisher

Sels, P., **T. Dewilde**, P. Vansteenwegen, and D. Cattrysse. (2011) "Automated, passenger time optimal, robust timetabling, using integer programming". In: *Proceedings of the 1st International Workshop on High-speed and Intercity Railways (IWHIR 2011)*. Ed. by Y.-Q. Ni and X.-W. Ye. Vol. 148. Lecture Notes in Electrical engineering. Springer-Verlag Berlin Heidelberg, pp. 87–92.

Papers at international scientific conferences and symposia, published in full in proceedings

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2013) "Exploiting the potential of the available capacity in railway station areas to improve the robustness". In: *Proceedings of eighth Triennial Symposium on Transportation Analysis (TRISTAN)*. San Pedro de Atacama, Chile.

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2013) "Robust railway station planning: An interaction between routing timetabling and platforming". In: *Proceedings of 5th International Conference on Railway Operations Modelling and Analysis (RailCopenhagen)*. Ed. by I. Hansen, O. Nielsen, A. Landex, J. Törnquist Krasemann, and J. Pachl. Copenhagen, Denmark. Young Railway Operations Research Award 2013 of the International Association of Railway Operations Research.

Sels, P., **T. Dewilde**, D. Cattrysse, and P. Vansteenwegen. (2013) "Expected passenger travel time for train schedule evaluation and optimization". In: *Proceedings of 5th International Conference on Railway Operations Modelling and Analysis (RailCopenhagen)*. Ed. by I. Hansen, O. Nielsen, A. Landex, J. Törnquist Krasemann, and J. Pachl. Copenhagen, Denmark.

Sels, P., **T. Dewilde**, D. Cattrysse, and P. Vansteenwegen. (2013) "A passenger knock-on delay model for timetable optimisation". In: *Proceedings of 3rd International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. Ed. by T. Albrecht, B. Jaekel, and M. Lehnert. Dresden, Germany, pp. 377–387.

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2012) "Solving the train routing problem to decrease knock-on delay at station level". In: *Proceedings of 10th International Conference on Applied Mathematical*

Optimization and Modelling (APMOD). Ed. by L. Suhl, G. Mitra, C. Lucas, A. Koberstein, and L. Beckmann. Paderborn, Germany, pp. 256–261.

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2011) "Defining robustness of a railway timetable". In: *Proceedings of 4th International Conference on Railway Operations Modelling and Analysis (RailRome)*. Rome, Italy.

Sels, P., **T. Dewilde**, D. Cattrysse, and P. Vansteenwegen. (2011) "Deriving all passenger flows in a railway network from ticket sales data". In: *Proceedings of 4th International Conference on Railway Operations Modelling and Analysis (RailRome)*. Rome, Italy.

Sels, P., B. Waquet, **T. Dewilde**, D. Cattrysse, and P. Vansteenwegen. (2011) "Calculation of realistic railway station capacity by platforming feasibility checks". In: *Proceedings of 2nd International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. Ed. by F. Viti, C. Tampère, and B. Immers. Leuven, Belgium.

Dewilde, T., D. Cattrysse, S. Coene, F.C.R. Spieksma, and P. Vansteenwegen. (2010) "Heuristics for the traveling repairman problem with profits". In: *Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS '10)*. Ed. by T. Erlebach and M. Lübbecke. Liverpool, United Kingdom, pp. 34–44.

Meeting abstracts, presented at international scientific conferences and symposia, published or not published in proceedings or journals

Dewilde, T., S. Burggraeve, T. Ameye, P. Sels, D. Cattrysse, and P. Vansteenwegen. (2014) "The impact of preprocessing for solving the train routing problem". In: *Proceedings of 28th Annual conference of the Belgian Operations Research Society (ORBEL)*. Mons, Belgium, pp. 110–111.

Sels, P., **T. Dewilde**, D. Cattrysse, and P. Vansteenwegen. (2014) "Automatically and quickly planning platform and route of trains in railway stations". In: *Proceedings of 20th Conference of the International Federation of Operational Research Societies (IFORS)*. Barcelona, Spain.

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2013) "Adapting the train schedule due to planned infrastructure unavailability in complex railway stations". In: *Proceedings of International Conference on Operations Research (OR2013)*. Rotterdam, the Netherlands, pp. 63-64.

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2012) "On the improvement of robustness in railway station areas". In: *Proceedings of 25th European Conference on Operational Research (EURO)*. Vilnius, Lithuania.

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2012) "Improving the robustness of the railway system in Brussels". In: *Proceedings of 26th Annual Conference of the Belgian Operations Research Society (ORBEL)*. Brussels, Belgium, pp. 55–56.

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2011) "Knock, Knock, Knock-on on railway networks". In: *Proceedings of 19th Conference of the International Federation of Operational Research Societies (IFORS)*. Melbourne, Australia, pp. 53.

Dewilde, T., P. Sels, D. Cattrysse, and P. Vansteenwegen. (2011) "Defining robustness of a railway timetable". In: *Proceedings of 25th Annual conference of the Belgian Operations Research Society (ORBEL)*. Ghent, Belgium, pp. 108–109.

Sels, P., **T. Dewilde**, D. Cattrysse, and P. Vansteenwegen. (2011) "Deriving all passenger flows in a railway network from ticket sales data". In: *Proceedings of 25th Annual conference of the Belgian Operations Research Society (ORBEL)*. Ghent, Belgium, pp. 110–111.

Other publications

Burggraeve, S., **T. Dewilde**, T. Ameye, P. Sels, and P. Vansteenwegen. (2014) "Definitie van robuustheid voor treinsystemen (Defining robustness of a railway timetable) (in Dutch)". Accepted in *Tijdschrift Vervoerswetenschap*.

Dewilde, T., P. Vansteenwegen, and D. Cattrysse. (2010) "Definitie van robuustheid voor een spoorwegdienstregeling (Defining robustness of a railway timetable) (in Dutch)". In: *Technical report, KU Leuven, Belgium*.

Awards

Young Railway Operations Research Award 2013 of the International Association of Railway Operations Research (IAROR): **Dewilde, T.**, P. Sels, D. Cattrysse, and P. Vansteenwegen. (2013) “Robust railway station planning: An interaction between routing timetabling and platforming”. In: *Proceedings of 5th International Conference on Railway Operations Modelling and Analysis (RailCopenhagen)*. Ed. by I. Hansen, O. Nielsen, A. Landex, J. Törnquist Krasemann, and J. Pachl. Copenhagen, Denmark.

ORBEL Award to the best student’s thesis on Operations Research: **Dewilde, T.** (2010) “The profit-based latency problem”. In: *Proceedings of 24th Annual conference of the Belgian Operations Research Society (ORBEL)*. Liège, Belgium.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF MECHANICAL ENGINEERING
CENTRE FOR INDUSTRIAL MANAGEMENT, TRAFFIC AND INFRASTRUCTURE
Celestijnenlaan 300 bus 2422
3001 LEUVEN
Thijs.Dewilde@kuleuven.be
http://www.cib.kuleuven.be